

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

 Σπύρος Γ. Ζυγούρης  
Καθηγητής Πληροφορικής

 **spzygouris@gmail.com**

You **Tube**



Spyros Georgios Zygoris

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

## Πίνακες

1.18

Να γραφεί αλγόριθμος που θα διαβάσει ένα μονοδιάστατο πίνακα  $N$  θέσεων

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.18

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα μονοδιάστατο πίνακα  $N$  θέσεων και θα δημιουργεί το είδωλό του.

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.18

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα μονοδιάστατο πίνακα  $N$  θέσεων και θα δημιουργεί το είδωλό του.

Δηλαδή, αν ο αρχικός πίνακας  $A=[a_1, a_2, a_3, \dots, a_{N-1}, a_N]$ , τότε να δημιουργηθεί ο

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.18

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα μονοδιάστατο πίνακα  $N$  θέσεων και θα δημιουργεί το είδωλό του.

Δηλαδή, αν ο αρχικός πίνακας  $A=[a_1, a_2, a_3, \dots, a_{N-1}, a_N]$ , τότε να δημιουργηθεί ο πίν

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.18

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα μονοδιάστατο πίνακα  $N$  θέσεων και θα δημιουργεί το είδωλό του.

Δηλαδή, αν ο αρχικός πίνακας  $A=[a_1, a_2, a_3, \dots, a_{N-1}, a_N]$ , τότε να δημιουργηθεί ο πίνακας  $B=[a_N, a_{N-1}, \dots, a_3, a_2, a_1]$ ,

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

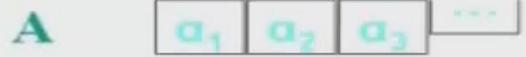
ΕΠΑΝΑΛΗΨΗ

1.18

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα μονοδιάστατο πίνακα  $N$  θέσεων και θα δημιουργεί το είδωλό του.

Δηλαδή, αν ο αρχικός πίνακας  $A=[a_1, a_2, a_3, \dots, a_{N-1}, a_N]$ , τότε να δημιουργηθεί ο πίνακας  $B=[a_N, a_{N-1}, \dots, a_3, a_2, a_1]$ .



## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.18

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα μονοδιάστατο πίνακα  $N$  θέσεων και θα δημιουργεί το είδωλό του.

Δηλαδή, αν ο αρχικός πίνακας  $A=[a_1, a_2, a_3, \dots, a_{N-1}, a_N]$ , τότε να δημιουργηθεί ο πίνακας  $B=[a_N, a_{N-1}, \dots, a_3, a_2, a_1]$ .

A 

$a_1$	$a_2$	$a_3$	...	...	...	$a_{N-1}$	$a_N$
-------	-------	-------	-----	-----	-----	-----------	-------

B 

--

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.18

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα μονοδιάστατο πίνακα  $N$  θέσεων και θα δημιουργεί το είδωλό του.

Δηλαδή, αν ο αρχικός πίνακας  $A=[a_1, a_2, a_3, \dots, a_{N-1}, a_N]$ , τότε να δημιουργηθεί ο πίνακας  $B=[a_N, a_{N-1}, \dots, a_3, a_2, a_1]$ ,



## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.18

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα μονοδιάστατο πίνακα  $N$  θέσεων και θα δημιουργεί το είδωλό του.

Δηλαδή, αν ο αρχικός πίνακας  $A=[a_1, a_2, a_3, \dots, a_{N-1}, a_N]$ , τότε να δημιουργηθεί ο πίνακας  $B=[a_N, a_{N-1}, \dots, a_3, a_2, a_1]$ ,

Παρατηρούμε ότι το στοιχείο  $A[1]$ ,

**A**

$a_1$	$a_2$	$a_3$	...	...	...	$a_{N-1}$	$a_N$
-------	-------	-------	-----	-----	-----	-----------	-------

**B**

$a_N$	$a_{N-1}$	...	...	...	$a_3$	$a_2$	$a_1$
-------	-----------	-----	-----	-----	-------	-------	-------

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.18

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα μονοδιάστατο πίνακα  $N$  θέσεων και θα δημιουργεί το είδωλό του.

Δηλαδή, αν ο αρχικός πίνακας  $A=[a_1, a_2, a_3, \dots, a_{N-1}, a_N]$ , τότε να δημιουργηθεί ο πίνακας  $B=[a_N, a_{N-1}, \dots, a_3, a_2, a_1]$ ,

[1]

A

$a_1$	$a_2$	$a_3$	...	...	...	$a_{N-1}$	$a_N$
-------	-------	-------	-----	-----	-----	-----------	-------

B

$a_N$	$a_{N-1}$	...	...	...	$a_3$	$a_2$	$a_1$
-------	-----------	-----	-----	-----	-------	-------	-------

Παρατηρούμε ότι το στοιχείο  $A[1]$ ,  
θα τοποθετηθεί στο B|

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.18

### Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα μονοδιάστατο πίνακα  $N$  θέσεων και θα δημιουργεί το είδωλό του.

Δηλαδή, αν ο αρχικός πίνακας  $A=[a_1, a_2, a_3, \dots, a_{N-1}, a_N]$ , τότε να δημιουργηθεί ο πίνακας  $B=[a_N, a_{N-1}, \dots, a_3, a_2, a_1]$ ,



Παρατηρούμε ότι το στοιχείο  $A[1]$ , θα τοποθετηθεί στο  $B[N]$ .

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

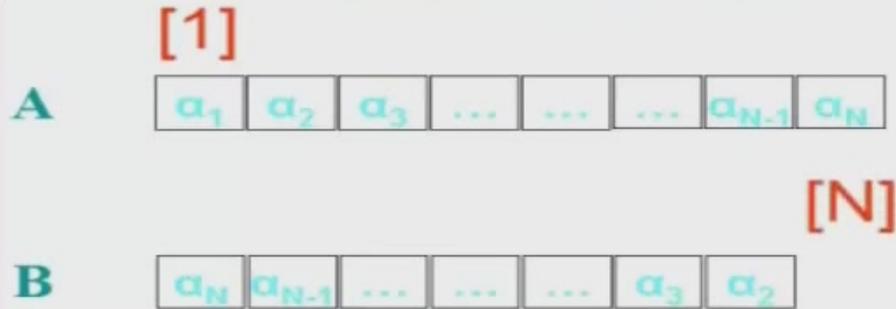
ΕΠΑΝΑΛΗΨΗ

1.18

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα μονοδιάστατο πίνακα  $N$  θέσεων και θα δημιουργεί το είδωλό του.

Δηλαδή, αν ο αρχικός πίνακας  $A=[a_1, a_2, a_3, \dots, a_{N-1}, a_N]$ , τότε να δημιουργηθεί ο πίνακας  $B=[a_N, a_{N-1}, \dots, a_3, a_2, a_1]$ ,



Παρατηρούμε ότι το στοιχείο  $A[1]$ , θα τοποθετηθεί στο  $B[N]$ , το  $A[2]$  στο  $B[N-1]$  κ

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

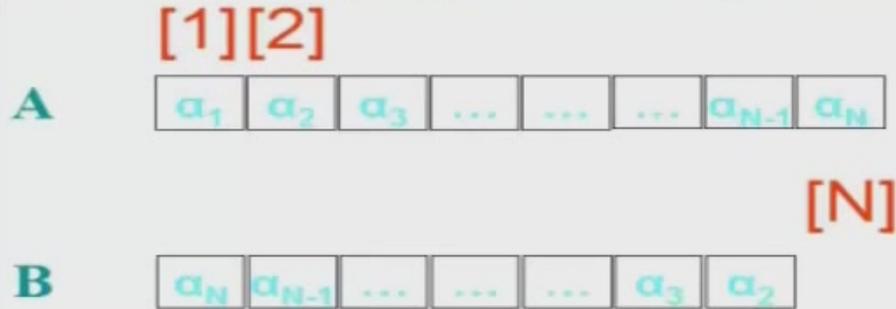
ΕΠΑΝΑΛΗΨΗ

1.18

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα μονοδιάστατο πίνακα  $N$  θέσεων και θα δημιουργεί το είδωλό του.

Δηλαδή, αν ο αρχικός πίνακας  $A=[a_1, a_2, a_3, \dots, a_{N-1}, a_N]$ , τότε να δημιουργηθεί ο πίνακας  $B=[a_N, a_{N-1}, \dots, a_3, a_2, a_1]$ ,



Παρατηρούμε ότι το στοιχείο  $A[1]$ , θα τοποθετηθεί στο  $B[N]$ , το  $A[2]$  στο  $B[N-1]$  κλπ.

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.18

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα μονοδιάστατο πίνακα  $N$  θέσεων και θα δημιουργεί το είδωλό του.

Δηλαδή, αν ο αρχικός πίνακας  $A=[a_1, a_2, a_3, \dots, a_{N-1}, a_N]$ , τότε να δημιουργηθεί ο πίνακας  $B=[a_N, a_{N-1}, \dots, a_3, a_2, a_1]$ ,

[1][2]

A

$a_1$	$a_2$	$a_3$	...	...	...	$a_{N-1}$	$a_N$
-------	-------	-------	-----	-----	-----	-----------	-------

[N-1][N]

B

$a_N$	$a_{N-1}$	...	...	...	$a_3$
-------	-----------	-----	-----	-----	-------

Παρατηρούμε ότι το στοιχείο  $A[1]$ , θα τοποθετηθεί στο  $B[N]$ , το  $A[2]$  στο  $B[N-1]$  κλπ.

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

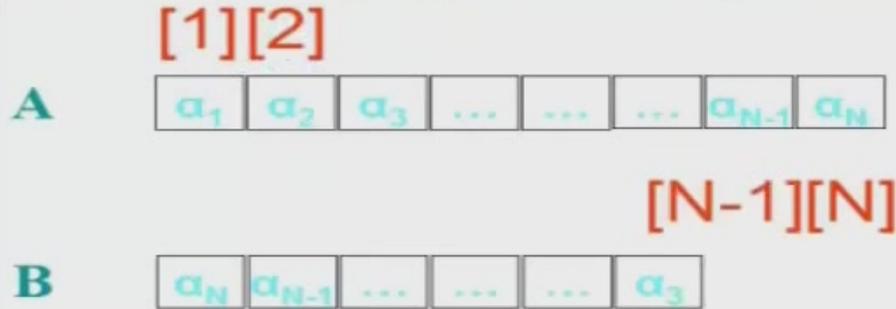
ΕΠΑΝΑΛΗΨΗ

1.18

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα μονοδιάστατο πίνακα  $N$  θέσεων και θα δημιουργεί το είδωλό του.

Δηλαδή, αν ο αρχικός πίνακας  $A=[a_1, a_2, a_3, \dots, a_{N-1}, a_N]$ , τότε να δημιουργηθεί ο πίνακας  $B=[a_N, a_{N-1}, \dots, a_3, a_2, a_1]$ ,



Παρατηρούμε ότι το στοιχείο  $A[1]$ , θα τοποθετηθεί στο  $B[N]$ , το  $A[2]$  στο  $B[N-1]$  κλπ.

Το στοιχείο  $A[i]$ , θα τοποθετηθεί στο

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

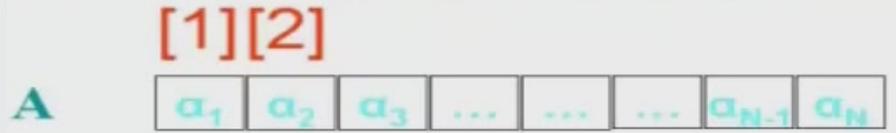
ΕΠΑΝΑΛΗΨΗ

1.18

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα μονοδιάστατο πίνακα  $N$  θέσεων και θα δημιουργεί το είδωλό του.

Δηλαδή, αν ο αρχικός πίνακας  $A=[a_1, a_2, a_3, \dots, a_{N-1}, a_N]$ , τότε να δημιουργηθεί ο πίνακας  $B=[a_N, a_{N-1}, \dots, a_3, a_2, a_1]$ ,



Παρατηρούμε ότι το στοιχείο  $A[1]$ , θα τοποθετηθεί στο  $B[N]$ , το  $A[2]$  στο  $B[N-1]$  κλπ.

Το στοιχείο  $A[i]$ , θα τοποθετηθεί στο  $B[N-i+1]$

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

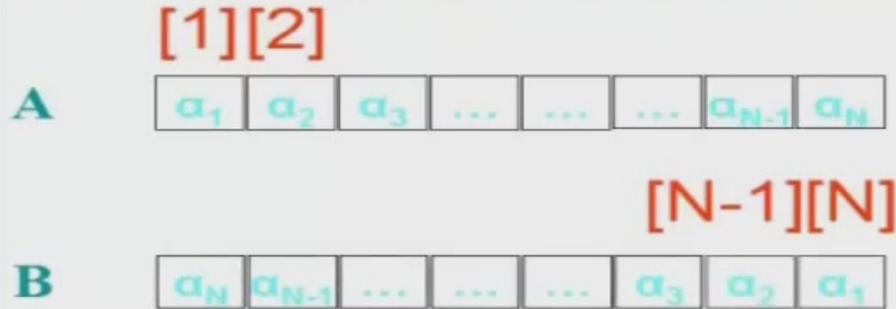
ΕΠΑΝΑΛΗΨΗ

1.18

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα μονοδιάστατο πίνακα  $N$  θέσεων και θα δημιουργεί το είδωλό του.

Δηλαδή, αν ο αρχικός πίνακας  $A=[a_1, a_2, a_3, \dots, a_{N-1}, a_N]$ , τότε να δημιουργηθεί ο πίνακας  $B=[a_N, a_{N-1}, \dots, a_3, a_2, a_1]$ ,



Παρατηρούμε ότι το στοιχείο  $A[1]$ , θα τοποθετηθεί στο  $B[N]$ , το  $A[2]$  στο  $B[N-1]$  κλπ.

Το στοιχείο  $A[i]$ , θα τοποθετηθεί στο  $B[N-i+1]$

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.18

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα μονοδιάστατο πίνακα  $N$  θέσεων και θα δημιουργεί το είδωλό του.

Δηλαδή, αν ο αρχικός πίνακας  $A=[a_1, a_2, a_3, \dots, a_{N-1}, a_N]$ , τότε να δημιουργηθεί ο πίνακας  $B=[a_N, a_{N-1}, \dots, a_3, a_2, a_1]$ ,



Παρατηρούμε ότι το στοιχείο  $A[1]$ , θα τοποθετηθεί στο  $B[N]$ , το  $A[2]$  στο  $B[N-1]$  κλπ.

Το στοιχείο  $A[i]$ , θα τοποθετηθεί στο  $B[N-i+1]$

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_Πίνακα
...
Τέλος Αντιστροφή_Πίνακα
    
```

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.18

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα μονοδιάστατο πίνακα  $N$  θέσεων και θα δημιουργεί το είδωλό του.

Δηλαδή, αν ο αρχικός πίνακας  $A=[a_1, a_2, a_3, \dots, a_{N-1}, a_N]$ , τότε να δημιουργηθεί ο πίνακας  $B=[a_N, a_{N-1}, \dots, a_3, a_2, a_1]$ ,

[1][2]

A 

$a_1$	$a_2$	$a_3$	...	...	...	$a_{N-1}$	$a_N$
-------	-------	-------	-----	-----	-----	-----------	-------

[N-1][N]

B 

$a_N$	$a_{N-1}$	...	...	...	$a_3$	$a_2$	$a_1$
-------	-----------	-----	-----	-----	-------	-------	-------

Παρατηρούμε ότι το στοιχείο  $A[1]$ , θα τοποθετηθεί στο  $B[N]$ , το  $A[2]$  στο  $B[N-1]$  κλπ.

Το στοιχείο  $A[i]$ , θα τοποθετηθεί στο  $B[N-i+1]$

Ο αλγόριθμος είναι ο ακόλουθος:

Αλγόριθμος Αντιστροφή\_Πίνακα  
Δεδομένα //  $N, A$

Τέλος Αντιστροφή\_Πίνακα

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.18

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα μονοδιάστατο πίνακα  $N$  θέσεων και θα δημιουργεί το είδωλό του.

Δηλαδή, αν ο αρχικός πίνακας  $A=[a_1, a_2, a_3, \dots, a_{N-1}, a_N]$ , τότε να δημιουργηθεί ο πίνακας  $B=[a_N, a_{N-1}, \dots, a_3, a_2, a_1]$ ,



Παρατηρούμε ότι το στοιχείο  $A[1]$ , θα τοποθετηθεί στο  $B[N]$ , το  $A[2]$  στο  $B[N-1]$  κλπ.

Το στοιχείο  $A[i]$ , θα τοποθετηθεί στο  $B[N-i+1]$

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_Πίνακα
Δεδομένα // N, A//

Τέλος Αντιστροφή_Πίνακα
    
```

Αρχή\_επανάληψης  
 Εμφάνισε "Δώσε το πλήθος των στοιχείων"  
 Διάβασε N  
 Μέχρις\_ότου  $N > 0$   
 Για i από 1 μέχρι N  
 Εμφάνισε "Δώσε το στοιχείο", i, "του πίνακα A"  
 Διάβασε A[i]  
 Τέλος\_επανάληψης

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

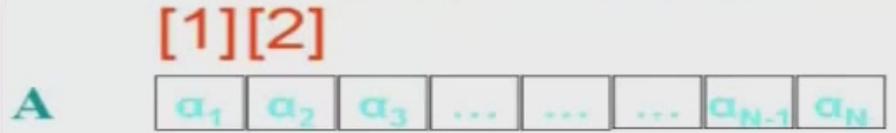
## ΕΠΑΝΑΛΗΨΗ

1.18

### Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα μονοδιάστατο πίνακα  $N$  θέσεων και θα δημιουργεί το είδωλό του.

Δηλαδή, αν ο αρχικός πίνακας  $A=[a_1, a_2, a_3, \dots, a_{N-1}, a_N]$ , τότε να δημιουργηθεί ο πίνακας  $B=[a_N, a_{N-1}, \dots, a_3, a_2, a_1]$ ,



Παρατηρούμε ότι το στοιχείο  $A[1]$ , θα τοποθετηθεί στο  $B[N]$ , το  $A[2]$  στο  $B[N-1]$  κλπ.

Το στοιχείο  $A[i]$ , θα τοποθετηθεί στο  $B[N-i+1]$

```

Αρχή_επανάληψης
  Εμφάνισε "Δώσε το πλήθος των στοιχείων"
  Διάβασε N
  Μέχρις_ότου N > 0
  Για i από 1 μέχρι N
    Εμφάνισε "Δώσε το στοιχείο", i, "του πίνακα A"
    Διάβασε A[i]
  Τέλος_επανάληψης
  
```

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_Πίνακα
  Δεδομένα // N, A//

  Τέλος Αντιστροφή_Πίνακα
  
```

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

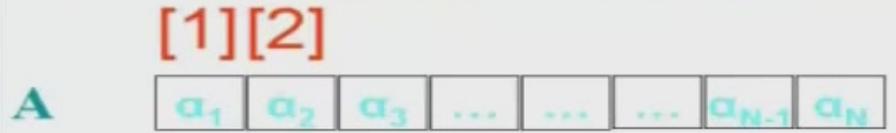
## ΕΠΑΝΑΛΗΨΗ

1.18

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα μονοδιάστατο πίνακα  $N$  θέσεων και θα δημιουργεί το είδωλό του.

Δηλαδή, αν ο αρχικός πίνακας  $A=[a_1, a_2, a_3, \dots, a_{N-1}, a_N]$ , τότε να δημιουργηθεί ο πίνακας  $B=[a_N, a_{N-1}, \dots, a_3, a_2, a_1]$ ,



Παρατηρούμε ότι το στοιχείο  $A[1]$ , θα τοποθετηθεί στο  $B[N]$ , το  $A[2]$  στο  $B[N-1]$  κλπ.

Το στοιχείο  $A[i]$ , θα τοποθετηθεί στο  $B[N-i+1]$

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_Πίνακα
Δεδομένα // N, A//

Για i από 1 μέχρι N

Τέλος Αντιστροφή_Πίνακα
    
```

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

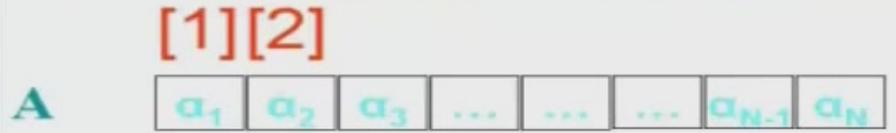
## ΕΠΑΝΑΛΗΨΗ

1.18

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα μονοδιάστατο πίνακα  $N$  θέσεων και θα δημιουργεί το είδωλό του.

Δηλαδή, αν ο αρχικός πίνακας  $A = [a_1, a_2, a_3, \dots, a_{N-1}, a_N]$ , τότε να δημιουργηθεί ο πίνακας  $B = [a_N, a_{N-1}, \dots, a_3, a_2, a_1]$ ,



Παρατηρούμε ότι το στοιχείο  $A[1]$ , θα τοποθετηθεί στο  $B[N]$ , το  $A[2]$  στο  $B[N-1]$  κλπ.

Το στοιχείο  $A[i]$ , θα τοποθετηθεί στο  $B[N-i+1]$

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_Πίνακα
Δεδομένα // N, A//

Για i από 1 μέχρι N
    B[i] ← A[N-i+1]

Τέλος Αντιστροφή_Πίνακα
    
```

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

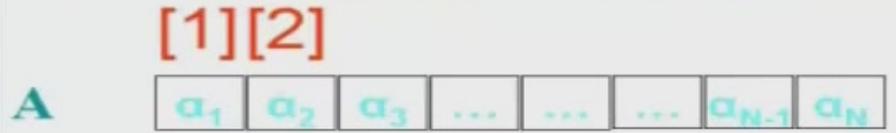
## ΕΠΑΝΑΛΗΨΗ

1.18

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα μονοδιάστατο πίνακα  $N$  θέσεων και θα δημιουργεί το είδωλό του.

Δηλαδή, αν ο αρχικός πίνακας  $A=[a_1, a_2, a_3, \dots, a_{N-1}, a_N]$ , τότε να δημιουργηθεί ο πίνακας  $B=[a_N, a_{N-1}, \dots, a_3, a_2, a_1]$ ,



Παρατηρούμε ότι το στοιχείο  $A[1]$ , θα τοποθετηθεί στο  $B[N]$ , το  $A[2]$  στο  $B[N-1]$  κλπ.

Το στοιχείο  $A[i]$ , θα τοποθετηθεί στο  $B[N-i+1]$

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_Πίνακα
Δεδομένα // N, A//

Για i από 1 μέχρι N
    B[i] ← A[N-i+1]

Τέλος Αντιστροφή_Πίνακα
    
```

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.18

### Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα μονοδιάστατο πίνακα  $N$  θέσεων και θα δημιουργεί το είδωλό του.

Δηλαδή, αν ο αρχικός πίνακας  $A=[a_1, a_2, a_3, \dots, a_{N-1}, a_N]$ , τότε να δημιουργηθεί ο πίνακας  $B=[a_N, a_{N-1}, \dots, a_3, a_2, a_1]$ ,

[1][2]

A

$a_1$	$a_2$	$a_3$	...	...	...	$a_{N-1}$	$a_N$
-------	-------	-------	-----	-----	-----	-----------	-------

[N-1][N]

B

$a_N$	$a_{N-1}$	...	...	...	$a_3$	$a_2$	$a_1$
-------	-----------	-----	-----	-----	-------	-------	-------

Παρατηρούμε ότι το στοιχείο  $A[1]$ , θα τοποθετηθεί στο  $B[N]$ , το  $A[2]$  στο  $B[N-1]$  κλπ.

Το στοιχείο  $A[i]$ , θα τοποθετηθεί στο  $B[N-i+1]$

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_Πίνακα
Δεδομένα // N, A//

Για i από 1 μέχρι N
    B[i] ← A[N-i+1]
Τέλος_επανάληψης

Τέλος Αντιστροφή_Πίνακα
    
```

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.18

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα μονοδιάστατο πίνακα  $N$  θέσεων και θα δημιουργεί το είδωλό του.

Δηλαδή, αν ο αρχικός πίνακας  $A=[a_1, a_2, a_3, \dots, a_{N-1}, a_N]$ , τότε να δημιουργηθεί ο πίνακας  $B=[a_N, a_{N-1}, \dots, a_3, a_2, a_1]$ ,

[1][2]

A

$a_1$	$a_2$	$a_3$	...	...	...	$a_{N-1}$	$a_N$
-------	-------	-------	-----	-----	-----	-----------	-------

[N-1][N]

B

$a_N$	$a_{N-1}$	...	...	...	$a_3$	$a_2$	$a_1$
-------	-----------	-----	-----	-----	-------	-------	-------

Παρατηρούμε ότι το στοιχείο  $A[1]$ , θα τοποθετηθεί στο  $B[N]$ , το  $A[2]$  στο  $B[N-1]$  κλπ.

Το στοιχείο  $A[i]$ , θα τοποθετηθεί στο  $B[N-i+1]$

Ο αλγόριθμος είναι ο ακόλουθος:

Αλγόριθμος Αντιστροφή\_Πίνακα

Δεδομένα //  $N, A$

Για  $i$  από 1 μέχρι  $N$

$B[i] \leftarrow A[N-i+1]$

Τέλος\_επανάληψης

Αποτελέσματα //  $N, B$

Τέλος Αντιστροφή\_Πίνακα

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.18

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα μονοδιάστατο πίνακα  $N$  θέσεων και θα δημιουργεί το είδωλό του.

Δηλαδή, αν ο αρχικός πίνακας  $A=[a_1, a_2, a_3, \dots, a_{N-1}, a_N]$ , τότε να δημιουργηθεί ο πίνακας  $B=[a_N, a_{N-1}, \dots, a_3, a_2, a_1]$ ,



Παρατηρούμε ότι το στοιχείο  $A[1]$ , θα τοποθετηθεί στο  $B[N]$ , το  $A[2]$  στο  $B[N-1]$  κλπ.

Το στοιχείο  $A[i]$ , θα τοποθετηθεί στο  $B[N-i+1]$

Ο αλγόριθμος είναι ο ακόλουθος:

Για  $i$  από 1 μέχρι  $N$   
 Εμφάνισε  $B[i]$   
 Τέλος\_επανάληψης

```

Αλγόριθμος Αντιστροφή_Πίνακα
Δεδομένα // N, A//

Για i από 1 μέχρι N
    B[i] ← A[N-i+1]
Τέλος_επανάληψης

Αποτελέσματα // N, B//

Τέλος Αντιστροφή_Πίνακα
    
```

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.18

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα μονοδιάστατο πίνακα  $N$  θέσεων και θα δημιουργεί το είδωλό του.

Δηλαδή, αν ο αρχικός πίνακας  $A=[a_1, a_2, a_3, \dots, a_{N-1}, a_N]$ , τότε να δημιουργηθεί ο πίνακας  $B=[a_N, a_{N-1}, \dots, a_3, a_2, a_1]$ ,



Παρατηρούμε ότι το στοιχείο  $A[1]$ , θα τοποθετηθεί στο  $B[N]$ , το  $A[2]$  στο  $B[N-1]$  κλπ.

Το στοιχείο  $A[i]$ , θα τοποθετηθεί στο  $B[N-i+1]$

Ο αλγόριθμος είναι ο ακόλουθος:

Για  $i$  από 1 μέχρι  $N$   
 Εμφάνισε  $B[i]$   
 Τέλος\_επανάληψης

```

Αλγόριθμος Αντιστροφή_Πίνακα
Δεδομένα // N, A//

Για i από 1 μέχρι N
    B[i] ← A[N-i+1]
Τέλος_επανάληψης

Αποτελέσματα // N, B//

Τέλος Αντιστροφή_Πίνακα
    
```

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.18

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα μονοδιάστατο πίνακα  $N$  θέσεων και θα δημιουργεί το είδωλό του.

Δηλαδή, αν ο αρχικός πίνακας  $A=[a_1, a_2, a_3, \dots, a_{N-1}, a_N]$ , τότε να δημιουργηθεί ο πίνακας  $B=[a_N, a_{N-1}, \dots, a_3, a_2, a_1]$ ,

[1][2]

A

$a_1$	$a_2$	$a_3$	...	...	...	$a_{N-1}$	$a_N$
-------	-------	-------	-----	-----	-----	-----------	-------

[N-1][N]

B

$a_N$	$a_{N-1}$	...	...	...	$a_3$	$a_2$	$a_1$
-------	-----------	-----	-----	-----	-------	-------	-------

Παρατηρούμε ότι το στοιχείο  $A[1]$ , θα τοποθετηθεί στο  $B[N]$ , το  $A[2]$  στο  $B[N-1]$  κλπ.

Το στοιχείο  $A[i]$ , θα τοποθετηθεί στο  $B[N-i+1]$

Ο αλγόριθμος είναι ο ακόλουθος:

Αλγόριθμος Αντιστροφή\_Πίνακα

Δεδομένα //  $N, A$ //Για  $i$  από 1 μέχρι  $N$  $B[i] \leftarrow A[N-i+1]$ 

Τέλος\_επανάληψης

Αποτελέσματα //  $N, B$ //

Τέλος Αντιστροφή\_Πίνακα

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

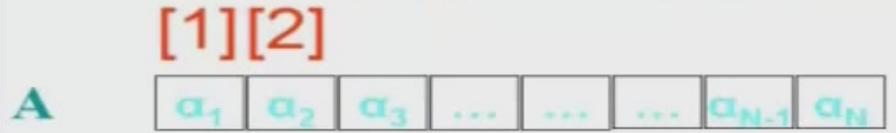
## ΕΠΑΝΑΛΗΨΗ

1.18

### Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα μονοδιάστατο πίνακα  $N$  θέσεων και θα δημιουργεί το είδωλό του.

Δηλαδή, αν ο αρχικός πίνακας  $A=[a_1, a_2, a_3, \dots, a_{N-1}, a_N]$ , τότε να δημιουργηθεί ο πίνακας  $B=[a_N, a_{N-1}, \dots, a_3, a_2, a_1]$ ,



Παρατηρούμε ότι το στοιχείο  $A[1]$ , θα τοποθετηθεί στο  $B[N]$ , το  $A[2]$  στο  $B[N-1]$  κλπ.

Το στοιχείο  $A[i]$ , θα τοποθετηθεί στο  $B[N-i+1]$

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_Πίνακα
Δεδομένα // N, A//

Για i από 1 μέχρι N
    B[i] ← A[N-i+1]
Τέλος_επανάληψης

Αποτελέσματα // N, B//

Τέλος Αντιστροφή_Πίνακα
    
```

Όταν θα μας ζητηθεί να δημιουργηθεί και :

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.18

### Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα μονοδιάστατο πίνακα  $N$  θέσεων και θα δημιουργεί το είδωλό του.

Δηλαδή, αν ο αρχικός πίνακας  $A=[a_1, a_2, a_3, \dots, a_{N-1}, a_N]$ , τότε να δημιουργηθεί ο πίνακας  $B=[a_N, a_{N-1}, \dots, a_3, a_2, a_1]$ ,



Παρατηρούμε ότι το στοιχείο  $A[1]$ , θα τοποθετηθεί στο  $B[N]$ , το  $A[2]$  στο  $B[N-1]$  κλπ.

Το στοιχείο  $A[i]$ , θα τοποθετηθεί στο  $B[N-i+1]$

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_Πίνακα
Δεδομένα // N, A//

Για i από 1 μέχρι N
    B[i] ← A[N-i+1]
Τέλος_επανάληψης

Αποτελέσματα // N, B//

Τέλος Αντιστροφή_Πίνακα
    
```

Όταν θα μας ζητηθεί να δημιουργηθεί και όχι να εμφανίσουμε τότε χρησιμοποιούμε τη εντολή εκ;

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.18

### Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα μονοδιάστατο πίνακα  $N$  θέσεων και θα δημιουργεί το είδωλό του.

Δηλαδή, αν ο αρχικός πίνακας  $A=[a_1, a_2, a_3, \dots, a_{N-1}, a_N]$ , τότε να δημιουργηθεί ο πίνακας  $B=[a_N, a_{N-1}, \dots, a_3, a_2, a_1]$ ,



Παρατηρούμε ότι το στοιχείο  $A[1]$ , θα τοποθετηθεί στο  $B[N]$ , το  $A[2]$  στο  $B[N-1]$  κλπ.

Το στοιχείο  $A[i]$ , θα τοποθετηθεί στο  $B[N-i+1]$

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_Πίνακα
Δεδομένα // N, A//
    Για i από 1 μέχρι N
        B[i] ← A[N-i+1]
    Τέλος_επανάληψης
Αποτελέσματα // N, B//
Τέλος Αντιστροφή_Πίνακα
    
```

Όταν θα μας ζητηθεί να δημιουργηθεί και όχι να εμφανίσουμε τότε χρησιμοποιούμε τη εντολή εκχώρησης. ←

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

## Πίνακες

1.19

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος

 $a_1$

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος

 $a_1$   $a_2$   $a_3$   $a_4$

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος

 $a_1$   $a_2$   $a_3$   $a_4$   $a_5$   $a_6$   $a_7$

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος

Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του

 $a_1$   $a_2$   $a_3$   $a_4$   $a_5$   $a_6$   $a_7$

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος

Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα

$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$
-------	-------	-------	-------	-------	-------	-------

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος

[1] [2] [3] [4] [5] [6] [7]

$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$
-------	-------	-------	-------	-------	-------	-------

Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα.

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

αντίστροφος

επταψήφιος

[1] [2] [3] [4] [5] [6] [7]

$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$
-------	-------	-------	-------	-------	-------	-------

Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα.

Στη συνέχεια θα εκτυπώσουμε τον πίνακα

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος

[1] [2] [3] [4] [5] [6] [7]

$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$
-------	-------	-------	-------	-------	-------	-------

Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα.

Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο

αντίστροφος

$a_7$
-------

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος

[1] [2] [3] [4] [5] [6] [7]

$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$
-------	-------	-------	-------	-------	-------	-------

αντίστροφος

$a_7$	$a_6$	$a_5$	$a_4$	$a_3$	$a_2$	$a_1$
-------	-------	-------	-------	-------	-------	-------

Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα.

Στη συνέχεια θα εκτυπώσουμε τον πίνακα

από το τελευταίο  
μέχρι το πρώτο.

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος

[1]	[2]	[3]	[4]	[5]	[6]	[7]
$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$

αντίστροφος

$a_7$	$a_6$	$a_5$	$a_4$	$a_3$	$a_2$	$a_1$
-------	-------	-------	-------	-------	-------	-------

Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη διάσπαση του αριθμού θα χρησιμοποιήσουμε τη

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος

[1]	[2]	[3]	[4]	[5]	[6]	[7]
$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$

αντίστροφος

$a_7$	$a_6$	$a_5$	$a_4$	$a_3$	$a_2$	$a_1$
-------	-------	-------	-------	-------	-------	-------

Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη διάσπαση του αριθμού θα χρησιμοποιήσουμε τη συνεχή διαίρεση με το 10: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ.

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος

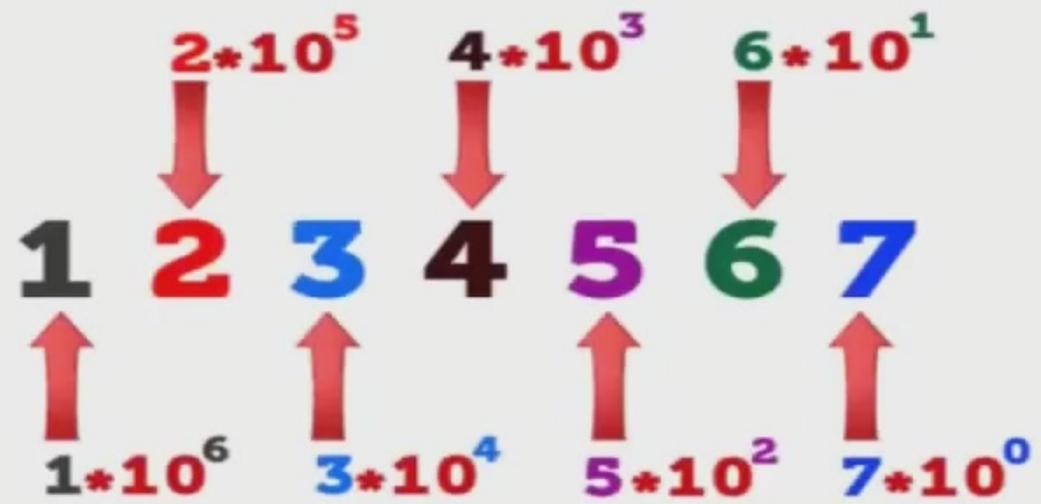
[1]	[2]	[3]	[4]	[5]	[6]	[7]
$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$

αντίστροφος

$a_7$	$a_6$	$a_5$	$a_4$	$a_3$	$a_2$	$a_1$
-------	-------	-------	-------	-------	-------	-------

Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη διάσπαση του αριθμού θα χρησιμοποιήσουμε τη συνεχή διαίρεση με το 10: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ.



# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος

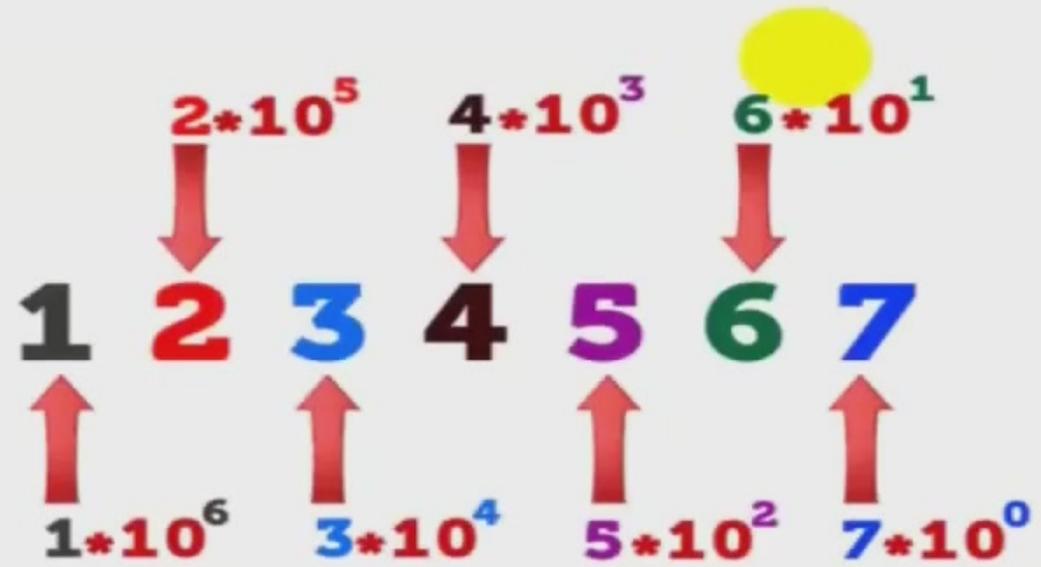
[1]	[2]	[3]	[4]	[5]	[6]	[7]
$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$

αντίστροφος

$a_7$	$a_6$	$a_5$	$a_4$	$a_3$	$a_2$	$a_1$
-------	-------	-------	-------	-------	-------	-------

Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη διάσπαση του αριθμού θα χρησιμοποιήσουμε τη συνεχή διαίρεση με το 10: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ.



# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.19

### Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος

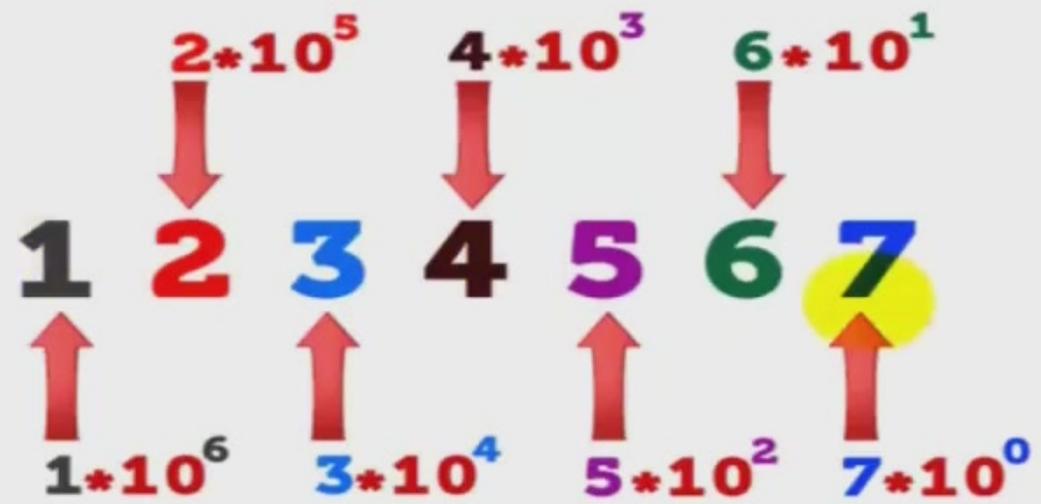
[1]	[2]	[3]	[4]	[5]	[6]	[7]
$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$

αντίστροφος

$a_7$	$a_6$	$a_5$	$a_4$	$a_3$	$a_2$	$a_1$
-------	-------	-------	-------	-------	-------	-------

Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη διάσπαση του αριθμού θα χρησιμοποιήσουμε τη συνεχή διαίρεση με το 10: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ.



# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.19

### Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος

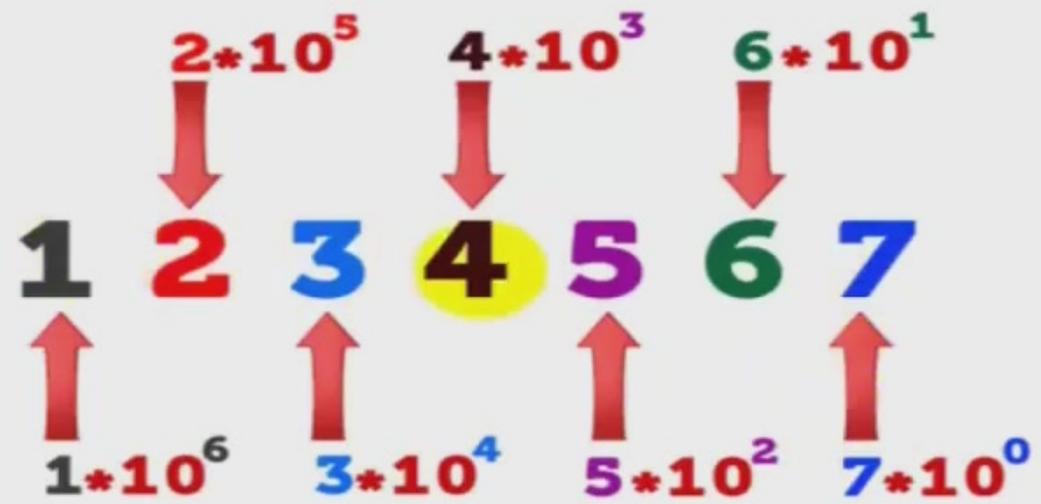
[1]	[2]	[3]	[4]	[5]	[6]	[7]
$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$

αντίστροφος

$a_7$	$a_6$	$a_5$	$a_4$	$a_3$	$a_2$	$a_1$
-------	-------	-------	-------	-------	-------	-------

Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη διάσπαση του αριθμού θα χρησιμοποιήσουμε τη συνεχή διαίρεση με το 10: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ.



# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.19

### Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος

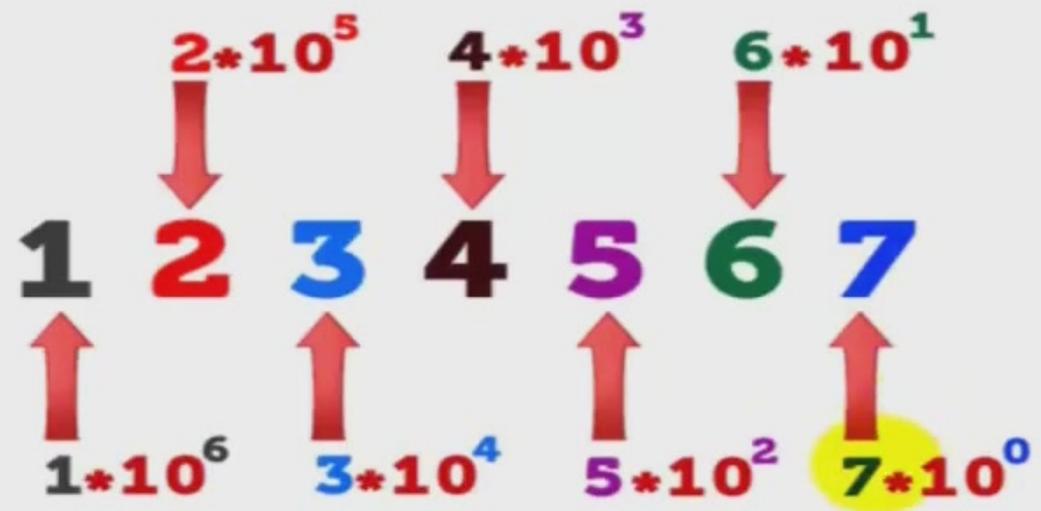
[1]	[2]	[3]	[4]	[5]	[6]	[7]
$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$

αντίστροφος

$a_7$	$a_6$	$a_5$	$a_4$	$a_3$	$a_2$	$a_1$
-------	-------	-------	-------	-------	-------	-------

Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη διάσπαση του αριθμού θα χρησιμοποιήσουμε τη συνεχή διαίρεση με το 10: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.



# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος

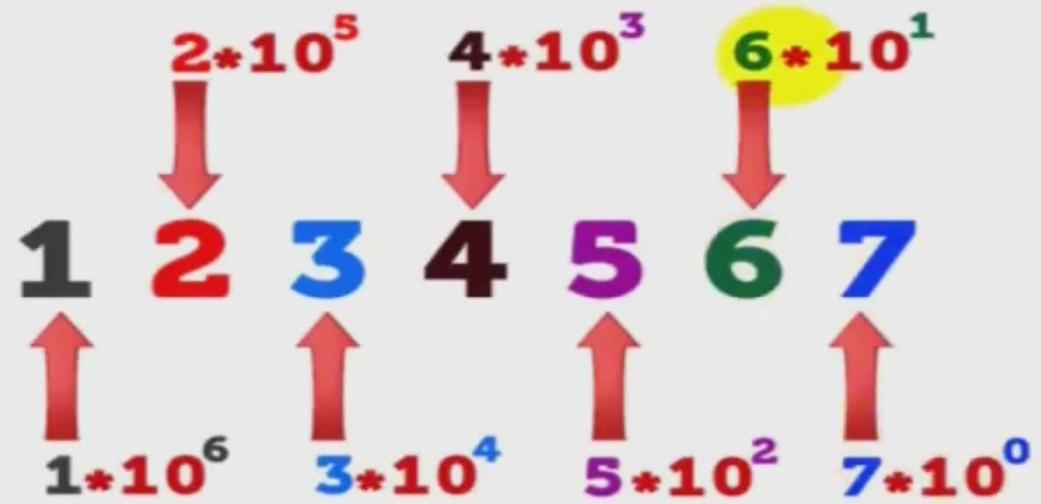
[1]	[2]	[3]	[4]	[5]	[6]	[7]
$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$

αντίστροφος

$a_7$	$a_6$	$a_5$	$a_4$	$a_3$	$a_2$	$a_1$
-------	-------	-------	-------	-------	-------	-------

Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη διάσπαση του αριθμού θα χρησιμοποιήσουμε τη συνεχή διαίρεση με το 10: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.



# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.19

### Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος

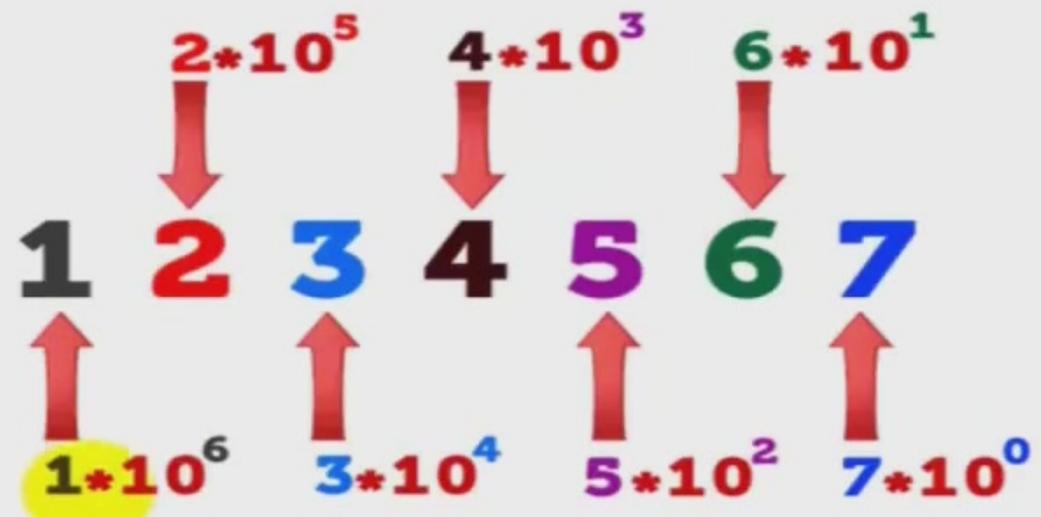
[1]	[2]	[3]	[4]	[5]	[6]	[7]
$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$

αντίστροφος

$a_7$	$a_6$	$a_5$	$a_4$	$a_3$	$a_2$	$a_1$
-------	-------	-------	-------	-------	-------	-------

Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη διάσπαση του αριθμού θα χρησιμοποιήσουμε τη συνεχή διαίρεση με το 10: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.



# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος

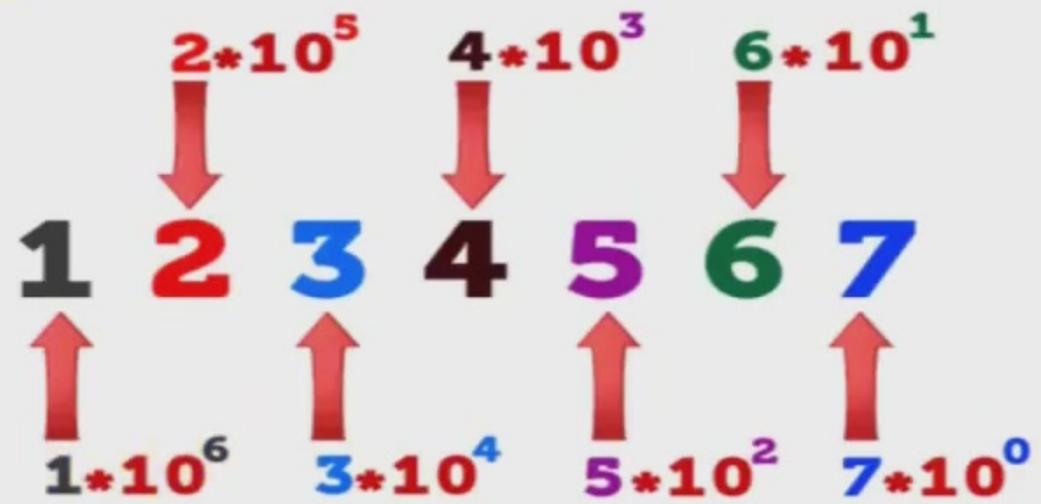
[1]	[2]	[3]	[4]	[5]	[6]	[7]
$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$

αντίστροφος

$a_7$	$a_6$	$a_5$	$a_4$	$a_3$	$a_2$	$a_1$
-------	-------	-------	-------	-------	-------	-------

Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη διάσπαση του αριθμού θα χρησιμοποιήσουμε τη συνεχή διαίρεση με το 10: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.



# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.19

### Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος

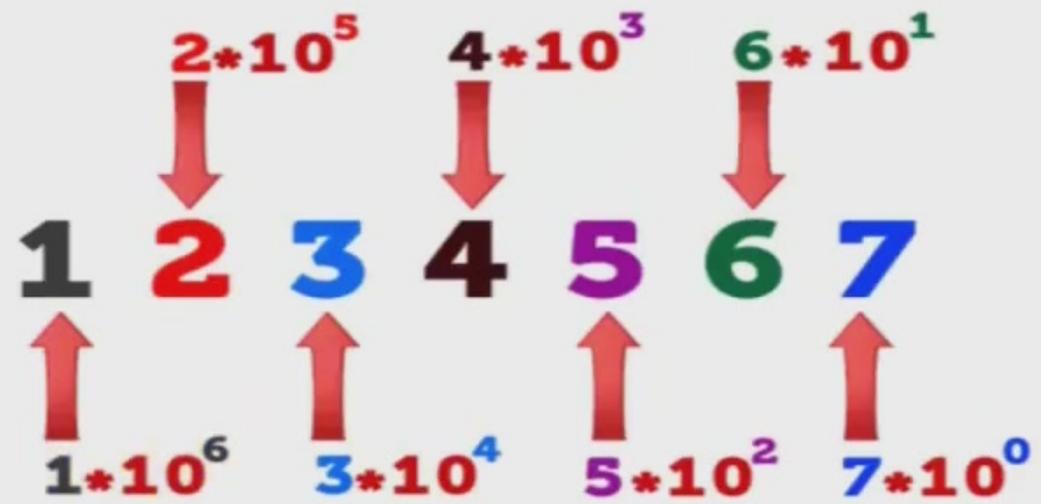
[1]	[2]	[3]	[4]	[5]	[6]	[7]
$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$

αντίστροφος

$a_7$	$a_6$	$a_5$	$a_4$	$a_3$	$a_2$	$a_1$
-------	-------	-------	-------	-------	-------	-------

Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη διάσπαση του αριθμού θα χρησιμοποιήσουμε τη συνεχή διαίρεση με το 10: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.



## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάσει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος

[1]	[2]	[3]	[4]	[5]	[6]	[7]
$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$

αντίστροφος

$a_7$	$a_6$	$a_5$	$a_4$	$a_3$	$a_2$	$a_1$
-------	-------	-------	-------	-------	-------	-------

Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:



# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

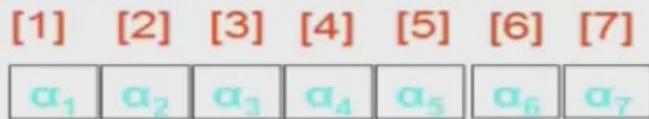
## ΕΠΑΝΑΛΗΨΗ

1.19

### Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

```

Αλγόριθμος Αντιστροφή_αριθμού
  Αρχή_επανάληψης
    Διαβασε αριθμός
  Τέλος Αντιστροφή_αριθμού
  
```

Ο αλγόριθμος είναι ο ακόλουθος:

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.19

### Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος

[1]	[2]	[3]	[4]	[5]	[6]	[7]
$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$

αντίστροφος

$a_7$	$a_6$	$a_5$	$a_4$	$a_3$	$a_2$	$a_1$
-------	-------	-------	-------	-------	-------	-------

Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
  Αρχή_επανάληψης
    Διαβασε αριθμός
    Μέχρις_ότου (αριθμός >= 1000000)
  Τέλος Αντιστροφή_αριθμού
  
```

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

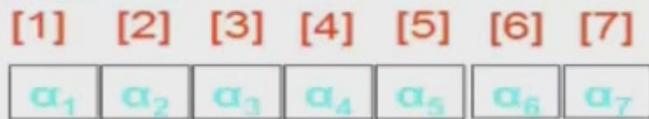
## ΕΠΑΝΑΛΗΨΗ

1.19

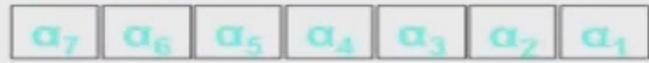
### Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

```

Αλγόριθμος Αντιστροφή_αριθμού
  Αρχή_επανάληψης
    Διαβασε αριθμός
    Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999)
      Τέλος Αντιστροφή_αριθμού
  
```

Ο αλγόριθμος είναι ο ακόλουθος:

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος

[1] [2] [3] [4] [5] [6] [7]

$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$
-------	-------	-------	-------	-------	-------	-------

αντίστροφος

$a_7$	$a_6$	$a_5$	$a_4$	$a_3$	$a_2$	$a_1$
-------	-------	-------	-------	-------	-------	-------

Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

```

Αλγόριθμος Αντιστροφή_αριθμού
  Αρχή_επανάληψης
    Διαβασε αριθμός
    Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999)
  
```

Ο αλγόριθμος είναι ο ακόλουθος:

```

Τέλος Αντιστροφή_αριθμού
  
```

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος

[1]	[2]	[3]	[4]	[5]	[6]	[7]
$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$

αντίστροφος

$a_7$	$a_6$	$a_5$	$a_4$	$a_3$	$a_2$	$a_1$
-------	-------	-------	-------	-------	-------	-------

Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

```

Αλγόριθμος Αντιστροφή_αριθμού
  Αρχή_επανάληψης
    Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999)

```

Ο αλγόριθμος είναι ο ακόλουθος:

```

Τέλος Αντιστροφή_αριθμού

```

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

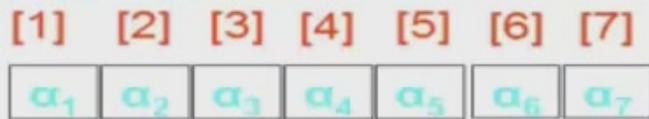
## ΕΠΑΝΑΛΗΨΗ

1.19

### Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός
  Τέλος

```

Ο αλγόριθμος είναι ο ακόλουθος:

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.19

### Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
  Αρχή_επανάληψης
    Διαβασε αριθμός
    Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
      βοηθητική ← αριθμός
  Τέλος Αντιστροφή_αριθμού
  
```

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

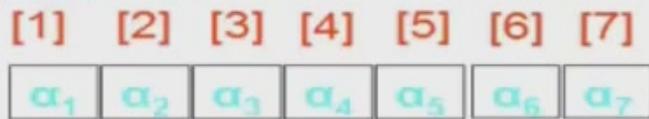
## ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
  Αρχή_επανάληψης
    Διαβασε αριθμός
    Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
      βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
    Για
      Τέλος Αντιστροφή_αριθμού
  
```

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

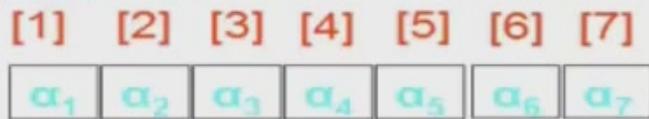
## ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
  Αρχή_επανάληψης
    Διαβασε αριθμός
    Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
      βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
      Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
        Τέλος Αντιστροφή_αριθμού
  
```

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

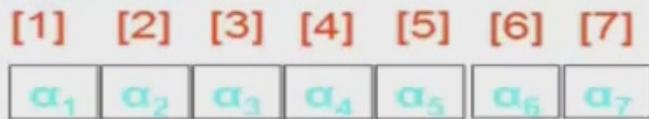
## ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
  Αρχή_επανάληψης
    Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
  Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
    ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
  Τέλος Αντιστροφή_αριθμού
  
```

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

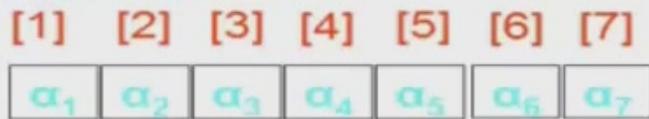
## ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
  Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
    ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
    ! Με την κατάλληλη διαίρεση
    ! πχ 7
  Τέλος Αντιστροφή_αριθμού
  
```

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.19

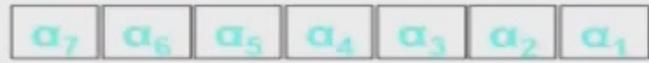
## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
  Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
    ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
    ! Με την κατάλληλη διαίρεση
    ! πχ 7601234 div 10^6 = 7
  
```

Τέλος Αντιστροφή\_αριθμού

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

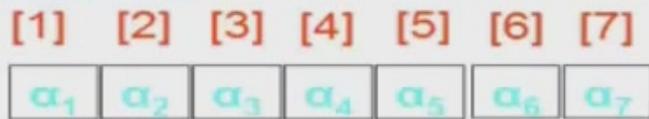
## ΕΠΑΝΑΛΗΨΗ

1.19

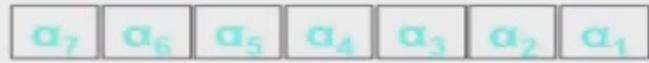
## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
  Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
    ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
    ! Με την κατάλληλη διαίρεση
    πίνακα[i] ← βοηθητική div 10^6 ! πχ 7601234 div 10^6 = 7
    βοηθητική ← βοηθητική mod 10^6
  Τέλος
  Αντιστροφή_αριθμού
  
```

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος

[1]	[2]	[3]	[4]	[5]	[6]	[7]
$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$

αντίστροφος

$a_7$	$a_6$	$a_5$	$a_4$	$a_3$	$a_2$	$a_1$
-------	-------	-------	-------	-------	-------	-------

Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
  Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
    ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
    ! Με την κατάλληλη διαίρεση
    ! πχ 7601234 div 10^6 = 7
  
```

Τέλος Αντιστροφή\_αριθμού

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

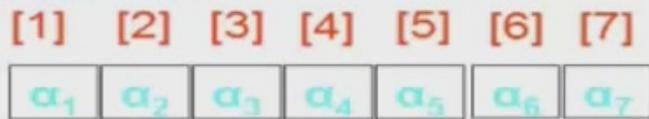
## ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
  Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
    ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
    ! Με την κατάλληλη διαίρεση
    ! πχ 7601234 div 10^6 = 7
  
```

Τέλος Αντιστροφή\_αριθμού

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

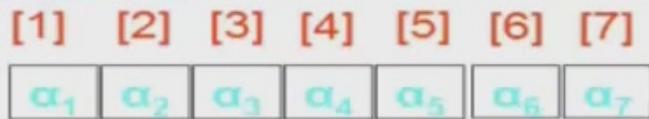
## ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
  Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
    ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
    ! Με την κατάλληλη διαίρεση
    A[i] ← βοηθητική div 10^(7-i) ! πχ 7601234 div 10^6 = 7
  Τέλος Αντιστροφή_αριθμού
  
```

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

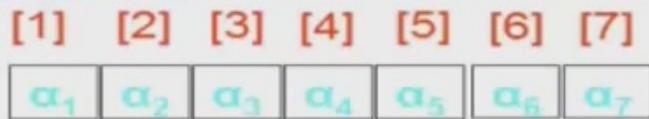
## ΕΠΑΝΑΛΗΨΗ

1.19

### Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
  Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
    ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
    ! Με την κατάλληλη διαίρεση
    A[i] ← βοηθητική div 10^(7-i) ! πχ 7601234 div 10^6 = 7
  Τέλος Αντιστροφή_αριθμού
  
```

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος

[1]	[2]	[3]	[4]	[5]	[6]	[7]
$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$

αντίστροφος

$a_7$	$a_6$	$a_5$	$a_4$	$a_3$	$a_2$	$a_1$
-------	-------	-------	-------	-------	-------	-------

Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
  Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
    ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
    ! Με την κατάλληλη διαίρεση
    A[i] ← βοηθητική div 10^(7-i) ! πχ 7601234 div 10^6 = 7
  Τέλος
  
```

Τέλος Αντιστροφή\_αριθμού

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
  Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
    ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
    ! Με την κατάλληλη διαίρεση
    A[i] ← βοηθητική div 10^(7-i) ! πχ 7601234 div 10^6 = 7
  Τέλος Αντιστροφή_αριθμού
  
```

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
  Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
    ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
    ! Με την κατάλληλη διαίρεση
    A[i] ← βοηθητική div 10^(7-i) ! πχ 7601234 div 10^6 = 7
  Τέλος Αντιστροφή_αριθμού
  
```

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

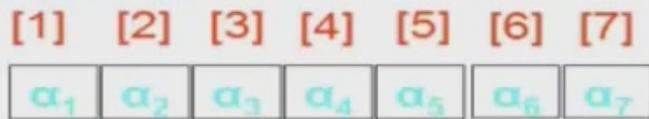
## ΕΠΑΝΑΛΗΨΗ

1.19

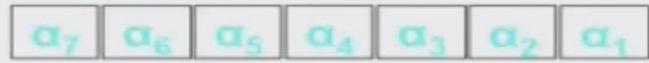
## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
  Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
    ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
    ! Με την κατάλληλη διαίρεση
    A[i] ← βοηθητική div 10^(7-i) ! πχ 7601234 div 10^6 = 7
  Τέλος Αντιστροφή_αριθμού
  
```

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος

[1]	[2]	[3]	[4]	[5]	[6]	[7]
$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$

αντίστροφος

$a_7$	$a_6$	$a_5$	$a_4$	$a_3$	$a_2$	$a_1$
-------	-------	-------	-------	-------	-------	-------

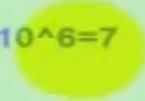
Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
  Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
    ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
    ! Με την κατάλληλη διαίρεση
    A[i] ← βοηθητική div 10^(7-i) ! πχ 7601234 div 10^6 = 7
  Τέλος
  Αντιστροφή_αριθμού
  
```



# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

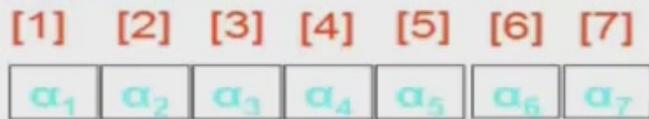
## ΕΠΑΝΑΛΗΨΗ

1.19

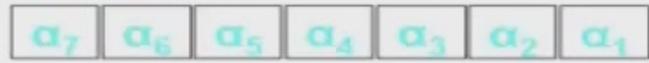
## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
  Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
    ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
    ! Με την κατάλληλη διαίρεση
    A[i] ← βοηθητική div 10^(7-i) ! πχ 7601234 div 10^6 = 7
  Τέλος Αντιστροφή_αριθμού
  
```

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

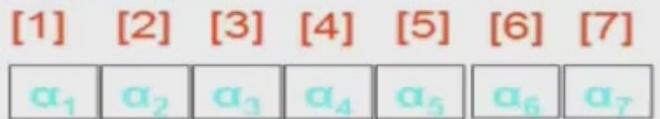
## ΕΠΑΝΑΛΗΨΗ

1.19

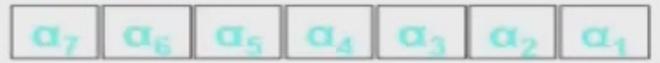
### Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
  Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
    ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
    ! Με την κατάλληλη διαίρεση
    A[i] ← βοηθητική div 10^(7-i) ! πχ 7601234 div 10^6 = 7
    βοηθητική ← βοηθητική mod 10^(7-i)
  Τέλος
Αντιστροφή_αριθμού
  
```

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

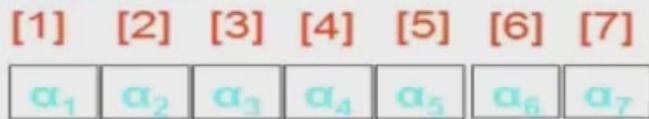
## ΕΠΑΝΑΛΗΨΗ

1.19

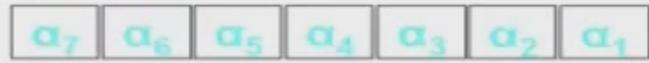
### Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
  Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
    ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
    ! Με την κατάλληλη διαίρεση
    A[i] ← βοηθητική div 10^(7-i) ! πχ 7601234 div 10^6 = 7
    βοηθητική ← βοηθητική mod 10^(7-i)
  Τέλος
Αντιστροφή_αριθμού
  
```

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

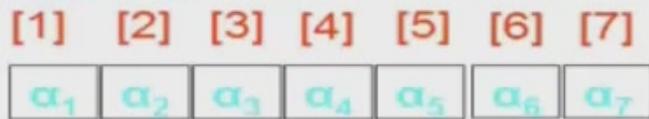
## ΕΠΑΝΑΛΗΨΗ

1.19

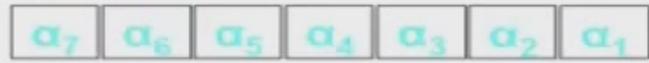
## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



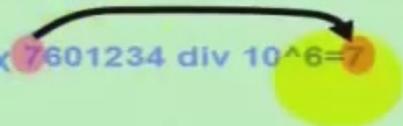
Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
  Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
    ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
    ! Με την κατάλληλη διαίρεση
    A[i] ← βοηθητική div 10^(7-i) ! πχ 7601234 div 10^6 = 7
    βοηθητική ← βοηθητική mod 10^(7-i)
  Τέλος Αντιστροφή_αριθμού
  
```



# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

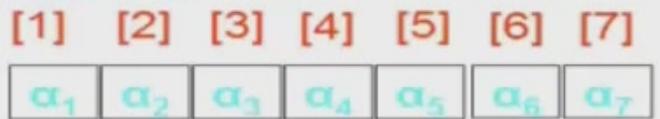
## ΕΠΑΝΑΛΗΨΗ

1.19

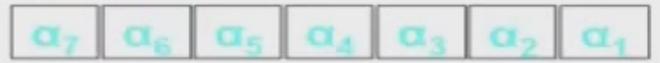
### Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



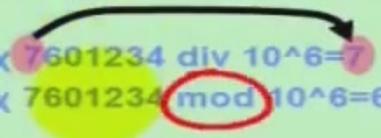
Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
  Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
    ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
    ! Με την κατάλληλη διαίρεση
    A[i] ← βοηθητική div 10^(7-i) ! πχ 7601234 div 10^6 = 7
    βοηθητική ← βοηθητική mod 10^(7-i) ! πχ 7601234 mod 10^6 = 601234
  Τέλος Αντιστροφή_αριθμού
  
```



# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

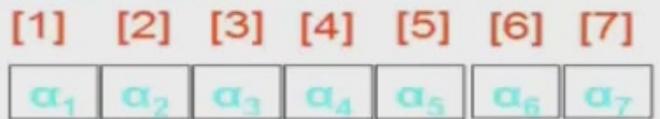
## ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



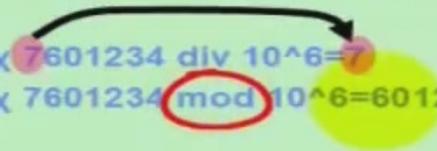
Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
  Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
    ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
    ! Με την κατάλληλη διαίρεση
    A[i] ← βοηθητική div 10^(7-i) ! πχ 7601234 div 10^6 = 7
    βοηθητική ← βοηθητική mod 10^(7-i) ! πχ 7601234 mod 10^6 = 601234
  Τέλος Αντιστροφή_αριθμού
  
```



# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

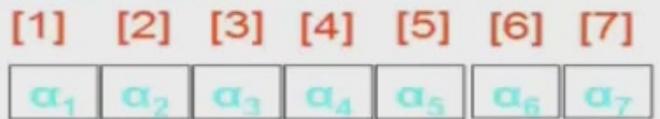
## ΕΠΑΝΑΛΗΨΗ

1.19

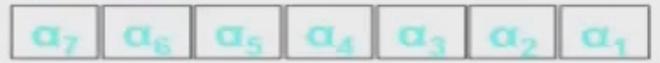
## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
  Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
    ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
    ! Με την κατάλληλη διαίρεση
    A[i] ← βοηθητική div 10^(7-i) ! πχ 7601234 div 10^6 = 7
    βοηθητική ← βοηθητική mod 10^(7-i) ! πχ 7601234 mod 10^6 = 601234
  Τέλος
Αντιστροφή_αριθμού
  
```

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

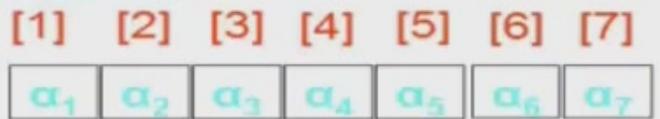
## ΕΠΑΝΑΛΗΨΗ

1.19

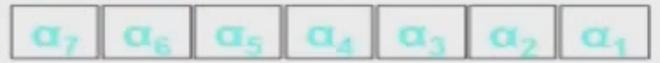
## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



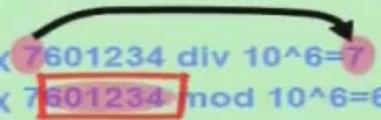
Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
  Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
    ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
    ! Με την κατάλληλη διαίρεση
    A[i] ← βοηθητική div 10^(7-i) ! πχ 7601234 div 10^6 = 7
    βοηθητική ← βοηθητική mod 10^(7-i) ! πχ 7601234 mod 10^6 = 601234
  Τέλος Αντιστροφή_αριθμού
  
```



# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

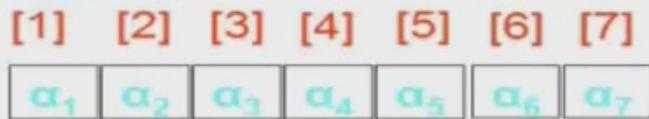
## ΕΠΑΝΑΛΗΨΗ

1.19

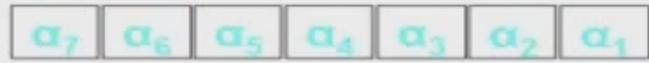
### Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



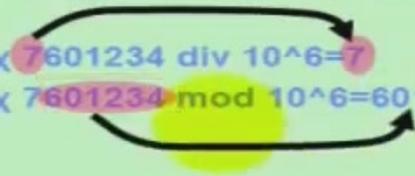
Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
  Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
    ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
    ! Με την κατάλληλη διαίρεση
    A[i] ← βοηθητική div 10^(7-i) ! πχ 7601234 div 10^6 = 7
    βοηθητική ← βοηθητική mod 10^(7-i) ! πχ 7601234 mod 10^6 = 601234
  Τέλος Αντιστροφή_αριθμού
  
```



# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.19

### Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
  Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
    ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
    ! Με την κατάλληλη διαίρεση
    A[i] ← βοηθητική div 10^(7-i) ! πχ 7601234 div 10^6 = 7
    βοηθητική ← βοηθητική mod 10^(7-i) ! πχ 7601234 mod 10^6 = 601234
  Τέλος Αντιστροφή_αριθμού
  
```



# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.19

### Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



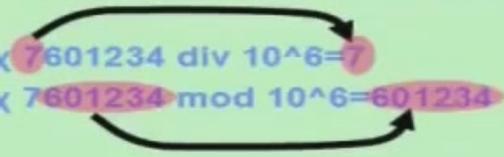
Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
  Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
    ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
    ! Με την κατάλληλη διαίρεση
    A[i] ← βοηθητική div 10^(7-i) ! πχ 7601234 div 10^6 = 7
    βοηθητική ← βοηθητική mod 10^(7-i) ! πχ 7601234 mod 10^6 = 601234
  Τέλος Αντιστροφή_αριθμού
  
```



# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.19

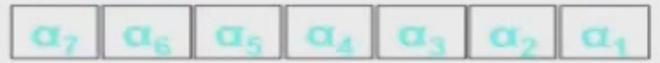
### Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



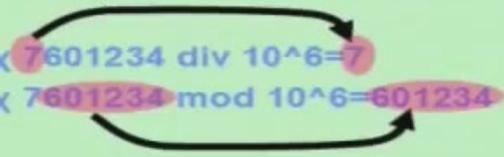
Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
  Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
    ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
    ! Με την κατάλληλη διαίρεση
    A[i] ← βοηθητική div 10^(7-i) ! πχ 7601234 div 10^6 = 7
    βοηθητική ← βοηθητική mod 10^(7-i) ! πχ 7601234 mod 10^6 = 601234
  Τέλος Αντιστροφή_αριθμού
  
```



# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.19

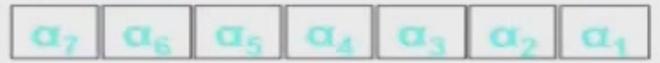
### Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



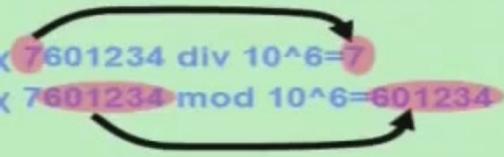
Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
  Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
    ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
    ! Με την κατάλληλη διαίρεση
    A[i] ← βοηθητική div 10^(7-i) ! πχ 7601234 div 10^6 = 7
    βοηθητική ← βοηθητική mod 10^(7-i) ! πχ 7601234 mod 10^6 = 601234
  Τέλος_επανάληψης
Τέλος Αντιστροφή_αριθμού
  
```



# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

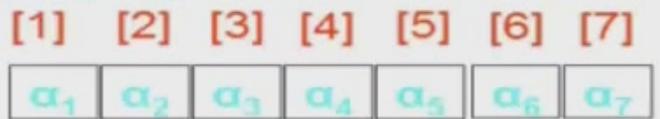
## ΕΠΑΝΑΛΗΨΗ

1.19

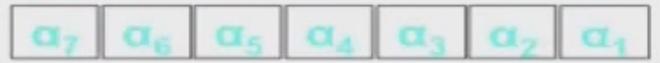
### Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



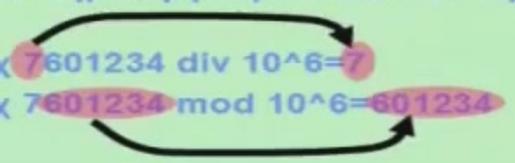
Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
  Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
    ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
    ! Με την κατάλληλη διαίρεση
    A[i] ← βοηθητική div 10^(7-i) ! πχ 7601234 div 10^6 = 7
    βοηθητική ← βοηθητική mod 10^(7-i) ! πχ 7601234 mod 10^6 = 601234
  Τέλος_επανάληψης
Γη
Τέλος Αντιστροφή_αριθμού
  
```



# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

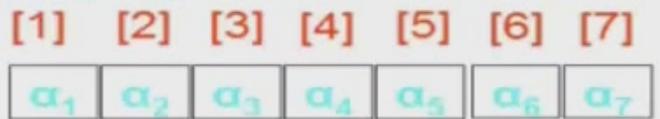
## ΕΠΑΝΑΛΗΨΗ

1.19

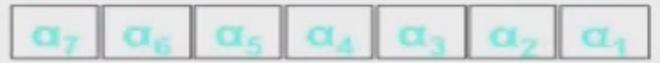
### Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



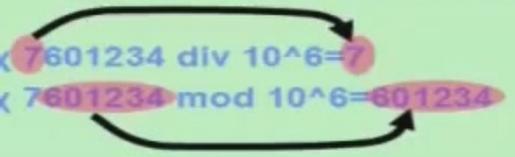
Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
  Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
    ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
    ! Με την κατάλληλη διαίρεση
    A[i] ← βοηθητική div 10^(7-i) ! πχ 7601234 div 10^6 = 7
    βοηθητική ← βοηθητική mod 10^(7-i) ! πχ 7601234 mod 10^6 = 601234
  Τέλος_επανάληψης
  Για i από 7 μέχρι 1 με_βήμα -1
  Τέλος Αντιστροφή_αριθμού
  
```



# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

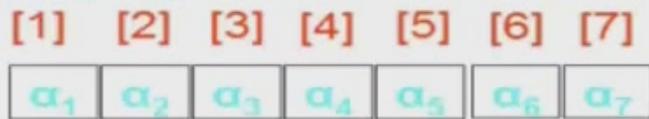
## ΕΠΑΝΑΛΗΨΗ

1.19

### Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



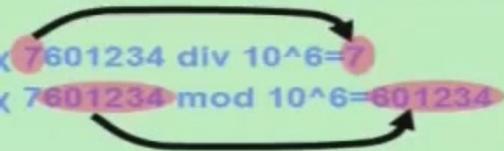
Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
  Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
    ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
    ! Με την κατάλληλη διαίρεση
    A[i] ← βοηθητική div 10^(7-i) ! πχ 7601234 div 10^6 = 7
    βοηθητική ← βοηθητική mod 10^(7-i) ! πχ 7601234 mod 10^6 = 601234
  Τέλος_επανάληψης
  Για i από 7 μέχρι 1 με_βήμα -1
  Τέλος Αντιστροφή_αριθμού
  
```



# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

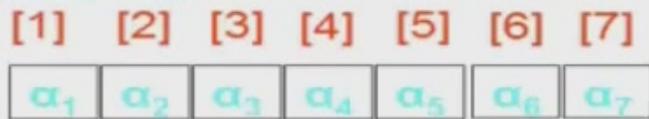
## ΕΠΑΝΑΛΗΨΗ

1.19

### Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

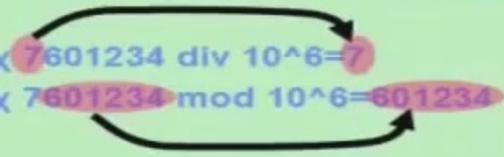
Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:



```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
    Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
      ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
      ! Με την κατάλληλη διαίρεση
      A[i] ← βοηθητική div 10^(7-i) ! πχ 7601234 div 10^6 = 7
      βοηθητική ← βοηθητική mod 10^(7-i) ! πχ 7601234 mod 10^6 = 601234
    Τέλος_επανάληψης
  Για i από 7 μέχρι 1 με_βήμα -1
  Τέλος Αντιστροφή_αριθμού
  
```



# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

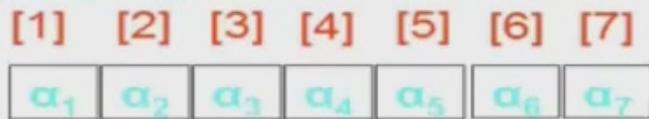
## ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

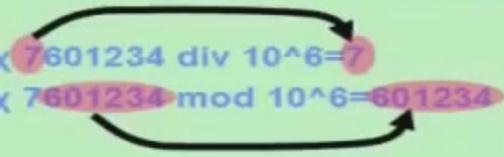
Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:



```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
  Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
    ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
    ! Με την κατάλληλη διαίρεση
    A[i] ← βοηθητική div 10^(7-i) ! πχ 7601234 div 10^6 = 7
    βοηθητική ← βοηθητική mod 10^(7-i) ! πχ 7601234 mod 10^6 = 601234
  Τέλος_επανάληψης
  Για i από 7 μέχρι 1 με_βήμα -1
    Γράψε A[i]
  Τέλος Αντιστροφή_αριθμού
  
```



# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

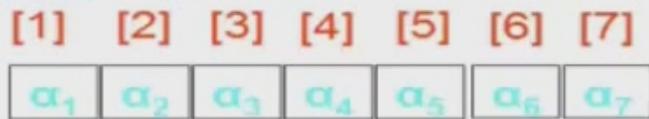
## ΕΠΑΝΑΛΗΨΗ

1.19

## Πίνακες

Να γραφεί αλγόριθμος που θα διαβάζει ένα επταψήφιο αριθμό και θα εκτυπώνει τα ψηφία του ανάποδα.

επταψήφιος



αντίστροφος



Θα πρέπει πρώτα να διαχωρίσουμε τα ψηφία του και να τα τοποθετήσουμε σε ένα πίνακα. Στη συνέχεια θα εκτυπώσουμε τον πίνακα από το τελευταίο μέχρι το πρώτο.

Για τη **διάσπαση του αριθμού** θα χρησιμοποιήσουμε τη συνεχή **διαίρεση με το 10**: Αρχικά με το  $10^6$ , στη συνέχεια  $10^5$  κλπ. Αποθηκεύοντας σε κάθε επανάληψη το υπόλοιπο της διαίρεσης.

Ο αλγόριθμος είναι ο ακόλουθος:



```

Αλγόριθμος Αντιστροφή_αριθμού
Αρχή_επανάληψης
  Διαβασε αριθμός
  Μέχρις_ότου (αριθμός >= 1000000) και (αριθμός <= 9999999) ! επταψήφιος
    βοηθητική ← αριθμός ! Κρατάμε αντίγραφο αριθμό
  Για i από 1 μέχρι 7 ! 7 νούμερα πρέπει να βρούμε
    ! Τοποθετούμε σε κάθε θέση του πίνακα, που δημιουργούμε το αντίστοιχο ψηφίο
    ! Με την κατάλληλη διαίρεση
    A[i] ← βοηθητική div 10^(7-i) ! πχ 7601234 div 10^6 = 7
    βοηθητική ← βοηθητική mod 10^(7-i) ! πχ 7601234 mod 10^6 = 601234
  Τέλος_επανάληψης
  Για i από 7 μέχρι 1 με_βήμα -1
    Γράψε A[i]
  Τέλος_επανάληψης
Τέλος Αντιστροφή_αριθμού
  
```

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

## Πίνακες

1.20

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αυτό το πρόβλημα δεν είναι υπολογιστικό αλλά απόφασης.

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αυτό το πρόβλημα δεν είναι υπολογιστικό αλλά απόφασης.

Για αυτά τα προβλήματα ενδείκνυται η  $\chi$

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αυτό το πρόβλημα δεν είναι υπολογιστικό αλλά απόφασης.

Για αυτά τα προβλήματα ενδείκνυται η χρήση λογικών μεταβλητών.

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αυτό το πρόβλημα δεν είναι υπολογιστικό αλλά απόφασης.

Για αυτά τα προβλήματα ενδείκνυται η χρήση λογικών μεταβλητών.

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αυτό το πρόβλημα δεν είναι υπολογιστικό αλλά απόφασης.

Για αυτά τα προβλήματα ενδείκνυται η χρήση λογικών μεταβλητών.

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αυτό το πρόβλημα δεν είναι υπολογιστικό αλλά απόφασης.

Για αυτά τα προβλήματα ενδείκνυται η χρήση λογικών μεταβλητών.

Αλη

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αυτό το πρόβλημα δεν είναι υπολογιστικό αλλά απόφασης.

Για αυτά τα προβλήματα ενδείκνυται η χρήση λογικών μεταβλητών.

Υποθέτουμε αρχικά ότι ισχύει η πρόταση της εκφώνησης.

Αληθές/Ψευδές  
Ναι/Όχι

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αυτό το πρόβλημα δεν είναι υπολογιστικό αλλά απόφασης.  
Για αυτά τα προβλήματα ενδείκνυται η χρήση λογικών μεταβλητών.  
Υποθέτουμε αρχικά ότι ισχύει η πρόταση της εκφώνησης.

Αληθές/Ψευδές  
Ναι/Όχι  
Αληθές

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αυτό το πρόβλημα δεν είναι υπολογιστικό αλλά απόφασης.  
Για αυτά τα προβλήματα ενδείκνυται η χρήση λογικών μεταβλητών.  
Υποθέτουμε αρχικά ότι ισχύει η πρόταση της εκφώνησης.  
Κατόπιν εξετάζουμε επαναληπτικά όλα τα στοιχεία και αν

Αληθές/Ψευδές  
Ναι/Όχι

τιμή ← Αληθές

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αυτό το πρόβλημα δεν είναι υπολογιστικό αλλά απόφασης.  
Για αυτά τα προβλήματα ενδείκνυται η χρήση λογικών μεταβλητών.  
Υποθέτουμε αρχικά ότι ισχύει η πρόταση της εκφώνησης.  
Κατόπιν εξετάζουμε επαναληπτικά όλα τα στοιχεία και αν  
εντοπίσουμε έστω και ένα που να μην ικανοποιεί την εκφώνηση

Αληθές/Ψευδές  
Ναι/Όχι

τιμή ← Αληθές

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αυτό το πρόβλημα δεν είναι υπολογιστικό αλλά απόφασης.  
Για αυτά τα προβλήματα ενδείκνυται η χρήση λογικών μεταβλητών.  
Υποθέτουμε αρχικά ότι ισχύει η πρόταση της εκφώνησης.  
Κατόπιν εξετάζουμε επαναληπτικά όλα τα στοιχεία και αν

εντοπίσουμε έστω και ένα που να μην ικανοποιεί την εκφώνηση  
τοποθετούμε την τιμή Ψευδής στη λογική μεταβλητή.

Αληθές/Ψευδές  
Ναι/Όχι

τιμή ← Αληθές

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αυτό το πρόβλημα δεν είναι υπολογιστικό αλλά απόφασης.

Για αυτά τα προβλήματα ενδείκνυται η χρήση λογικών μεταβλητών.

Υποθέτουμε αρχικά ότι ισχύει η πρόταση της εκφώνησης.

Κατόπιν εξετάζουμε επαναληπτικά όλα τα στοιχεία και αν

εντοπίσουμε έστω και ένα που να μην ικανοποιεί την εκφώνηση

τοποθετούμε την τιμή Ψευδής στη λογική μεταβλητή.

Αληθές/Ψευδές  
Ναι/Όχι

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αυτό το πρόβλημα δεν είναι υπολογιστικό αλλά απόφασης.  
Για αυτά τα προβλήματα ενδείκνυται η χρήση λογικών μεταβλητών.  
Υποθέτουμε αρχικά ότι ισχύει η πρόταση της εκφώνησης.  
Κατόπιν εξετάζουμε επαναληπτικά όλα τα στοιχεία και αν εντοπίσουμε έστω και ένα που να μην ικανοποιεί την εκφώνηση τοποθετούμε την τιμή Ψευδής στη λογική μεταβλητή.  
Τελικά ελέγχουμε την τιμή λογικής μεταβλητής.

Αληθές/Ψευδές  
Ναι/Όχι

τιμή ← Ψευδής

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αυτό το πρόβλημα δεν είναι υπολογιστικό αλλά απόφασης.  
Για αυτά τα προβλήματα ενδείκνυται η χρήση λογικών μεταβλητών.  
Υποθέτουμε αρχικά ότι ισχύει η πρόταση της εκφώνησης.  
Κατόπιν εξετάζουμε επαναληπτικά όλα τα στοιχεία και αν εντοπίσουμε έστω και ένα που να μην ικανοποιεί την εκφώνηση τοποθετούμε την τιμή Ψευδής στη λογική μεταβλητή.

Τελικά ελέγχουμε την τιμή λογικής μεταβλητής.

Στην άσκηση αυτή εξετάζουμε όλα τα αντιδιαμέτρικα στοιχεία

Αληθές/Ψευδές  
Ναι/Όχι

τιμή ← Ψευδής

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αυτό το πρόβλημα δεν είναι υπολογιστικό αλλά απόφασης.  
Για αυτά τα προβλήματα ενδείκνυται η χρήση λογικών μεταβλητών.  
Υποθέτουμε αρχικά ότι ισχύει η πρόταση της εκφώνησης.  
Κατόπιν εξετάζουμε επαναληπτικά όλα τα στοιχεία και αν εντοπίσουμε έστω και ένα που να μην ικανοποιεί την εκφώνηση τοποθετούμε την τιμή Ψευδής στη λογική μεταβλητή.  
Τελικά ελέγχουμε την τιμή λογικής μεταβλητής.

Στην άσκηση αυτή εξετάζουμε όλα τα αντιδιαμέτρικα στοιχεία και αν εντοπιστεί έστω και ένα ζευγάρι άνισων τιμών,

Αληθές/Ψευδές  
Ναι/Όχι

τιμή ← Ψευδής

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αυτό το πρόβλημα δεν είναι υπολογιστικό αλλά απόφασης.  
Για αυτά τα προβλήματα ενδείκνυται η χρήση λογικών μεταβλητών.  
Υποθέτουμε αρχικά ότι ισχύει η πρόταση της εκφώνησης.  
Κατόπιν εξετάζουμε επαναληπτικά όλα τα στοιχεία και αν εντοπίσουμε έστω και ένα που να μην ικανοποιεί την εκφώνηση τοποθετούμε την τιμή Ψευδής στη λογική μεταβλητή.  
Τελικά ελέγχουμε την τιμή λογικής μεταβλητής.  
Στην άσκηση αυτή εξετάζουμε όλα τα αντιδιαμέτρικα στοιχεία και αν εντοπιστεί έστω και ένα ζευγάρι άνισων τιμών,

Αληθές/Ψευδές  
Ναι/Όχι

τιμή ← Ψευδής

$a_1$	$a_2$	$a_3$	...	...	...	$a_{N-1}$	$a_N$
-------	-------	-------	-----	-----	-----	-----------	-------

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αυτό το πρόβλημα δεν είναι υπολογιστικό αλλά απόφασης.  
Για αυτά τα προβλήματα ενδείκνυται η χρήση λογικών μεταβλητών.  
Υποθέτουμε αρχικά ότι ισχύει η πρόταση της εκφώνησης.  
Κατόπιν εξετάζουμε επαναληπτικά όλα τα στοιχεία και αν εντοπίσουμε έστω και ένα που να μην ικανοποιεί την εκφώνηση τοποθετούμε την τιμή Ψευδής στη λογική μεταβλητή.  
Τελικά ελέγχουμε την τιμή λογικής μεταβλητής.  
Στην άσκηση αυτή εξετάζουμε όλα τα αντιδιαμέτρικα στοιχεία και αν εντοπιστεί έστω και ένα ζευγάρι άνισων τιμών,

Αληθές/Ψευδές  
Ναι/Όχι

τιμή ← Αληθές

τιμή ← Ψευδής

$a_1$	$a_2$	$a_3$	...	...	...	$a_{N-1}$	$a_N$
-------	-------	-------	-----	-----	-----	-----------	-------

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αυτό το πρόβλημα δεν είναι υπολογιστικό αλλά απόφασης.  
Για αυτά τα προβλήματα ενδείκνυται η χρήση λογικών μεταβλητών.  
Υποθέτουμε αρχικά ότι ισχύει η πρόταση της εκφώνησης.  
Κατόπιν εξετάζουμε επαναληπτικά όλα τα στοιχεία και αν εντοπίσουμε έστω και ένα που να μην ικανοποιεί την εκφώνηση τοποθετούμε την τιμή Ψευδής στη λογική μεταβλητή.  
Τελικά ελέγχουμε την τιμή λογικής μεταβλητής.  
Στην άσκηση αυτή εξετάζουμε όλα τα αντιδιαμέτρικα στοιχεία και αν εντοπιστεί έστω και ένα ζευγάρι άνισων τιμών, τοποθετούμε την τιμή ψευδής στη λογική μεταβλητή.

Αληθές/Ψευδές  
Ναι/Όχι

τιμή ← Αληθές

τιμή ← Ψευδής

$a_1$	$a_2$	$a_3$	...	...	...	$a_{N-1}$	$a_N$
-------	-------	-------	-----	-----	-----	-----------	-------

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αυτό το πρόβλημα δεν είναι υπολογιστικό αλλά απόφασης.  
Για αυτά τα προβλήματα ενδείκνυται η χρήση λογικών μεταβλητών.  
Υποθέτουμε αρχικά ότι ισχύει η πρόταση της εκφώνησης.  
Κατόπιν εξετάζουμε επαναληπτικά όλα τα στοιχεία και αν εντοπίσουμε έστω και ένα που να μην ικανοποιεί την εκφώνηση τοποθετούμε την τιμή Ψευδής στη λογική μεταβλητή.  
Τελικά ελέγχουμε την τιμή λογικής μεταβλητής.  
Στην άσκηση αυτή εξετάζουμε όλα τα αντιδιαμέτρικα στοιχεία και αν εντοπιστεί έστω και ένα ζευγάρι άνισων τιμών, τοποθετούμε την τιμή ψευδής στη λογική μεταβλητή.

Αληθές/Ψευδές  
Ναι/Όχι

τιμή ← Αληθές

τιμή ← Ψευδής

$a_1$	$a_2$	$a_3$	...	...	...	$a_{N-1}$	$a_N$
-------	-------	-------	-----	-----	-----	-----------	-------

Το αντιδιαμετρικό στοιχείο του  $A[1]$  είναι το  $A[N]$

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αυτό το πρόβλημα δεν είναι υπολογιστικό αλλά απόφασης.  
Για αυτά τα προβλήματα ενδείκνυται η χρήση λογικών μεταβλητών.  
Υποθέτουμε αρχικά ότι ισχύει η πρόταση της εκφώνησης.  
Κατόπιν εξετάζουμε επαναληπτικά όλα τα στοιχεία και αν εντοπίσουμε έστω και ένα που να μην ικανοποιεί την εκφώνηση τοποθετούμε την τιμή Ψευδής στη λογική μεταβλητή.  
Τελικά ελέγχουμε την τιμή λογικής μεταβλητής.  
Στην άσκηση αυτή εξετάζουμε όλα τα αντιδιαμέτρικα στοιχεία και αν εντοπιστεί έστω και ένα ζευγάρι άνισων τιμών, τοποθετούμε την τιμή ψευδής στη λογική μεταβλητή.

Αληθές/Ψευδές  
Ναι/Όχι

τιμή ← Αληθές

τιμή ← Ψευδής

$a_1$	$a_2$	$a_3$	...	...	...	$a_{N-1}$	$a_N$
-------	-------	-------	-----	-----	-----	-----------	-------

Το αντιδιαμετρικό στοιχείο του  $A[1]$  είναι το  $A[N]$

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αυτό το πρόβλημα δεν είναι υπολογιστικό αλλά απόφασης.  
Για αυτά τα προβλήματα ενδείκνυται η χρήση λογικών μεταβλητών.  
Υποθέτουμε αρχικά ότι ισχύει η πρόταση της εκφώνησης.  
Κατόπιν εξετάζουμε επαναληπτικά όλα τα στοιχεία και αν εντοπίσουμε έστω και ένα που να μην ικανοποιεί την εκφώνηση τοποθετούμε την τιμή Ψευδής στη λογική μεταβλητή.  
Τελικά ελέγχουμε την τιμή λογικής μεταβλητής.  
Στην άσκηση αυτή εξετάζουμε όλα τα αντιδιαμετρικά στοιχεία και αν εντοπιστεί έστω και ένα ζευγάρι άνισων τιμών, τοποθετούμε την τιμή ψευδής στη λογική μεταβλητή.

Αληθές/Ψευδές  
Ναι/Όχι

τιμή ← Αληθές

τιμή ← Ψευδής

$a_1$	$a_2$	$a_3$	...	...	...	$a_{N-1}$	$a_N$
-------	-------	-------	-----	-----	-----	-----------	-------

Το αντιδιαμετρικό στοιχείο του  $A[1]$  είναι το  $A[N]$ Το αντιδιαμετρικό στοιχείο του  $A[2]$  είναι το  $A[N-1]$

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αυτό το πρόβλημα δεν είναι υπολογιστικό αλλά απόφασης.  
Για αυτά τα προβλήματα ενδείκνυται η χρήση λογικών μεταβλητών.  
Υποθέτουμε αρχικά ότι ισχύει η πρόταση της εκφώνησης.  
Κατόπιν εξετάζουμε επαναληπτικά όλα τα στοιχεία και αν εντοπίσουμε έστω και ένα που να μην ικανοποιεί την εκφώνηση τοποθετούμε την τιμή Ψευδής στη λογική μεταβλητή.  
Τελικά ελέγχουμε την τιμή λογικής μεταβλητής.  
Στην άσκηση αυτή εξετάζουμε όλα τα αντιδιαμετρικά στοιχεία και αν εντοπιστεί έστω και ένα ζευγάρι άνισων τιμών, τοποθετούμε την τιμή ψευδής στη λογική μεταβλητή.

Αληθές/Ψευδές  
Ναι/Όχι

τιμή ← Αληθές

τιμή ← Ψευδής

$a_1$	$a_2$	$a_3$	...	...	...	$a_{N-1}$	$a_N$
-------	-------	-------	-----	-----	-----	-----------	-------

Το αντιδιαμετρικό στοιχείο του  $A[1]$  είναι το  $A[N]$

Το αντιδιαμετρικό στοιχείο του  $A[2]$  είναι το  $A[N-1]$

Το αντιδιαμετρικό στοιχείο του  $A[i]$  είναι το  $A[N-i+1]$

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αυτό το πρόβλημα δεν είναι υπολογιστικό αλλά απόφασης.  
Για αυτά τα προβλήματα ενδείκνυται η χρήση λογικών μεταβλητών.  
Υποθέτουμε αρχικά ότι ισχύει η πρόταση της εκφώνησης.  
Κατόπιν εξετάζουμε επαναληπτικά όλα τα στοιχεία και αν εντοπίσουμε έστω και ένα που να μην ικανοποιεί την εκφώνηση τοποθετούμε την τιμή Ψευδής στη λογική μεταβλητή.  
Τελικά ελέγχουμε την τιμή λογικής μεταβλητής.  
Στην άσκηση αυτή εξετάζουμε όλα τα αντιδιαμετρικά στοιχεία και αν εντοπιστεί έστω και ένα ζευγάρι άνισων τιμών, τοποθετούμε την τιμή ψευδής στη λογική μεταβλητή.

Αληθές/Ψευδές  
Ναι/Όχι

τιμή ← Αληθές

τιμή ← Ψευδής

$a_1$	$a_2$	$a_3$	...	...	...	$a_{N-1}$	$a_N$
-------	-------	-------	-----	-----	-----	-----------	-------

Το αντιδιαμετρικό στοιχείο του  $A[1]$  είναι το  $A[N]$

Το αντιδιαμετρικό στοιχείο του  $A[2]$  είναι το  $A[N-1]$

Το αντιδιαμετρικό στοιχείο του  $A[i]$  είναι το  $A[N-i+1]$

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αλγόριθμος Συμμετρικός\_Πίνακας

Τέλος Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αλγόριθμος Συμμετρικός\_Πίνακας

Δεδομένα //  $N, A$ 

```
Αρχη_επιλοχωσης
Εμφάνιση "Δώστε το πλήθος των στοιχείων"
Διάβασε N
Μέχρις_ότου N>0
Για I από 1 μέχρι N
    Εμφάνισε "Δώστε το στοιχείο", I, "του πίνακα A"
    Διάβασε A[I]
Τέλος_επιλοχωσης
```



Τέλος Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αλγόριθμος Συμμετρικός\_Πίνακας

Δεδομένα //  $N, A$

συμμετρικός ← Αληθής

Τέλος Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αλγόριθμος Συμμετρικός\_Πίνακας

Δεδομένα //  $N, A$ 

συμμετρικός ← Αληθής

Τέλος Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αλγόριθμος Συμμετρικός\_Πίνακας

Δεδομένα //  $N, A$

συμμετρικός ← Αληθής ! Έστω πίνακας συμμετρικός

Τέλος Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αλγόριθμος Συμμετρικός\_Πίνακας

Δεδομένα //  $N, A$

συμμετρικός ← Αληθής ! Έστω πίνακας συμμετρικός

Για  $i$  από 1 μέχρι

Τέλος Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αλγόριθμος Συμμετρικός\_Πίνακας

Δεδομένα //  $N, A$

συμμετρικός ← Αληθής ! Έστω πίνακας συμμετρικός

Για  $i$  από 1 μέχρι  $(N \text{ div } 2)$

Τέλος Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αλγόριθμος Συμμετρικός\_Πίνακας

Δεδομένα //  $N, A$

συμμετρικός ← Αληθής ! Έστω πίνακας συμμετρικός

Για  $i$  από 1 μέχρι  $(N \text{ div } 2)$  ! Έλεγχος μέχρι το μέσο του πίνακα

Τέλος Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αλγόριθμος Συμμετρικός\_Πίνακας

Δεδομένα //  $N, A$  //

συμμετρικός  $\leftarrow$  Αληθής ! Έστω πίνακας συμμετρικός

Για  $i$  από 1 μέχρι  $(N \text{ div } 2)$  ! Έλεγχος μέχρι το μέσο του πίνακα

Αν  $A[i] \neq A[N+1-i]$  τότε

Τέλος Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αλγόριθμος Συμμετρικός\_Πίνακας

Δεδομένα //  $N, A$

συμμετρικός ← Αληθής ! Έστω πίνακας συμμετρικός

Για  $i$  από 1 μέχρι  $(N \text{ div } 2)$  ! Έλεγχος μέχρι το μέσο του πίνακα

Αν  $A[i] \neq A[N+1-i]$  τότε

Τέλος Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας

**Δεδομένα** //  $N, A$

συμμετρικός  $\leftarrow$  Αληθής ! Έστω πίνακας συμμετρικός

**Για**  $i$  από 1 μέχρι  $(N \text{ div } 2)$  ! Έλεγχος μέχρι το μέσο του πίνακα

**Αν**  $A[i] \neq A[N+1-i]$  **τότε** ! Αν βρεθεί έστω ένα ζευγάρι ανισων τιμών, τότε ο πίνακας δεν είναι συμμετρικός

**Τέλος** Συμμετρικός\_Πίνακας



## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας**Δεδομένα** //  $N, A$ 

συμμετρικός ← Αληθής ! Έστω πίνακας συμμετρικός

**Για**  $i$  από 1 μέχρι  $(N \text{ div } 2)$  ! Έλεγχος μέχρι το μέσο του πίνακα**Αν**  $A[i] \neq A[N+1-i]$  **τότε** ! Αν βρεθεί έστω ένα ζευγάρι άνισων τιμών, τότε ο πίνακας δεν είναι συμμετρικός**Τέλος** Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας**Δεδομένα** //  $N, A$ συμμετρικός  $\leftarrow$  Αληθής ! Έστω πίνακας συμμετρικός**Για**  $i$  από 1 μέχρι  $(N \text{ div } 2)$  ! Έλεγχος μέχρι το μέσο του πίνακα**Αν**  $A[i] \neq A[N+1-i]$  **τότε** ! Αν βρεθεί έστω ένα ζευγάρι άνισων τιμών, τότε ο πίνακας δεν είναι συμμετρικόςσυμμετρικός  $\leftarrow$  Ψευδής**Τέλος** Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αλγόριθμος Συμμετρικός\_Πίνακας

Δεδομένα //  $N, A$

συμμετρικός ← Αληθής ! Έστω πίνακας συμμετρικός

Για  $i$  από 1 μέχρι  $(N \text{ div } 2)$  ! Έλεγχος μέχρι το μέσο του πίνακα

Αν  $A[i] <> A[N+1-i]$  τότε ! Αν βρεθεί έστω ένα ζευγάρι άνισων τιμών ,τότε ο πίνακας δεν είναι συμμετρικός

συμμετρικός ← Ψευδής

Τέλος\_αν

Τέλος\_επανάληψης

Ρ

Τέλος Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αλγόριθμος Συμμετρικός\_Πίνακας

Δεδομένα //  $N, A$ 

συμμετρικός ← Αληθής ! Έστω πίνακας συμμετρικός

Για  $i$  από 1 μέχρι  $(N \text{ div } 2)$  ! Έλεγχος μέχρι το μέσο του πίνακαΑν  $A[i] \neq A[N+1-i]$  τότε ! Αν βρεθεί έστω ένα ζευγάρι άνισων τιμών ,τότε ο πίνακας δεν είναι συμμετρικός

συμμετρικός ← Ψευδής

Τέλος\_αν

Τέλος\_επανάληψης

Αν συμμετρικός = Αληθής τότε

Γράψε

Τέλος Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας**Δεδομένα** //  $N, A$ συμμετρικός  $\leftarrow$  Αληθής ! Έστω πίνακας συμμετρικός**Για**  $i$  από 1 μέχρι  $(N \text{ div } 2)$  ! Έλεγχος μέχρι το μέσο του πίνακα**Αν**  $A[i] \neq A[N+1-i]$  **τότε** ! Αν βρεθεί έστω ένα ζευγάρι άνισων τιμών, τότε ο πίνακας δεν είναι συμμετρικόςσυμμετρικός  $\leftarrow$  Ψευδής

Τέλος\_αν

Τέλος\_επανάληψης

**Αν** συμμετρικός = Αληθής **τότε**

Γράψε "Συμμετρικός πίνακας"

Αλλιώς

**Τέλος** Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας

**Δεδομένα** //  $N, A$

συμμετρικός  $\leftarrow$  Αληθής ! Έστω πίνακας συμμετρικός

**Για**  $i$  από 1 μέχρι  $(N \text{ div } 2)$  ! Έλεγχος μέχρι το μέσο του πίνακα

**Αν**  $A[i] \neq A[N+1-i]$  **τότε** ! Αν βρεθεί έστω ένα ζευγάρι άνισων τιμών, τότε ο πίνακας δεν είναι συμμετρικός

συμμετρικός  $\leftarrow$  Ψευδής

Τέλος\_αν

Τέλος\_επανάληψης

**Αν** συμμετρικός = Αληθής **τότε**

Γράψε "Συμμετρικός πίνακας"

Αλλιώς

Γράψε "Όχι Συμμετρικός πίνακας"

Τέλος\_αν

**Τέλος** Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας

**Δεδομένα** //  $N, A$

συμμετρικός  $\leftarrow$  Αληθής ! Έστω πίνακας συμμετρικός

**Για**  $i$  από 1 μέχρι  $(N \text{ div } 2)$  ! Έλεγχος μέχρι το μέσο του πίνακα

**Αν**  $A[i] \neq A[N+1-i]$  **τότε** ! Αν βρεθεί έστω ένα ζευγάρι άνισων τιμών ,τότε ο πίνακας δεν είναι συμμετρικός

συμμετρικός  $\leftarrow$  Ψευδής

Τέλος\_αν

Τέλος\_επανάληψης

**Αν** συμμετρικός = Αληθής **τότε**

Γράψε "Συμμετρικός πίνακας"

Αλλιώς

Γράψε "Όχι Συμμετρικός πίνακας"

Τέλος\_αν

**Τέλος** Συμμετρικός\_Πίνακας

Παρατηρού

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας

**Δεδομένα** //  $N, A$

συμμετρικός ← Αληθής ! Έστω πίνακας συμμετρικός

**Για**  $i$  από 1 μέχρι  $(N \text{ div } 2)$  ! Έλεγχος μέχρι το μέσο του πίνακα

**Αν**  $A[i] \neq A[N+1-i]$  τότε ! Αν βρεθεί έστω ένα ζευγάρι άνισων τιμών, τότε ο πίνακας δεν είναι συμμετρικός

συμμετρικός ← Ψευδής

Τέλος\_αν

Τέλος\_επανάληψης

**Αν** συμμετρικός = Αληθής τότε

Γράψε "Συμμετρικός πίνακας"

Αλλιώς

Γράψε "Όχι Συμμετρικός πίνακας"

Τέλος\_αν

**Τέλος** Συμμετρικός\_Πίνακας

Παρατηρούμε ότι αν βρεθεί κάποιο ζευγάρι άνισων τιμών

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αλγόριθμος Συμμετρικός\_Πίνακας

Δεδομένα //  $N, A$ συμμετρικός  $\leftarrow$  Αληθής ! Έστω πίνακας συμμετρικόςΓια  $i$  από 1 μέχρι  $(N \text{ div } 2)$  ! Έλεγχος μέχρι το μέσο του πίνακαΑν  $A[i] <> A[N+1-i]$  τότε ! Αν βρεθεί έστω ένα ζευγάρι άνισων τιμών, τότε ο πίνακας δεν είναι συμμετρικόςσυμμετρικός  $\leftarrow$  Ψευδής

Τέλος\_αν

Τέλος\_επανάληψης

Αν συμμετρικός = Αληθής τότε

Γράψε "Συμμετρικός πίνακας"

Αλλιώς

Γράψε "Όχι Συμμετρικός πίνακας"

Τέλος\_αν

Τέλος Συμμετρικός\_Πίνακας

Παρατηρούμε ότι αν βρεθεί κάποιο ζευγάρι άνισων τιμών

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αλγόριθμος Συμμετρικός\_Πίνακας

Δεδομένα //  $N, A$ 

συμμετρικός ← Αληθής ! Έστω πίνακας συμμετρικός

Για  $i$  από 1 μέχρι  $(N \text{ div } 2)$  ! Έλεγχος μέχρι το μέσο του πίνακαΑν  $A[i] \neq A[N+1-i]$  τότε ! Αν βρεθεί έστω ένα ζευγάρι άνισων τιμών, τότε ο πίνακας δεν είναι συμμετρικός

συμμετρικός ← Ψευδής

Τέλος αν

Τέλος επανάληψης

Αν συμμετρικός = Αληθής τότε

Γράψε "Συμμετρικός πίνακας"

Αλλιώς

Γράψε "Όχι Συμμετρικός πίνακας"

Τέλος\_αν

Τέλος Συμμετρικός\_Πίνακας

Παρατηρούμε ότι αν βρεθεί κάποιο ζευγάρι άνισων τιμών

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αλγόριθμος Συμμετρικός\_Πίνακας

Δεδομένα //  $N, A$ 

συμμετρικός ← Αληθής ! Έστω πίνακας συμμετρικός

Για  $i$  από 1 μέχρι  $(N \text{ div } 2)$  ! Έλεγχος μέχρι το μέσο του πίνακα

Αν  $A[i] <> A[N+1-i]$  τότε ! Αν βρεθεί έστω ένα ζευγάρι άνισων τιμών ,τότε ο πίνακας δεν είναι συμμετρικός

συμμετρικός ← Ψευδής

Τέλος\_αν

Τέλος\_επανάληψης

Αν συμμετρικός = Αληθής τότε

Γράψε "Συμμετρικός πίνακας"

Αλλιώς

Γράψε "Όχι Συμμετρικός πίνακας"

Τέλος\_αν

Τέλος Συμμετρικός\_Πίνακας

Παρατηρούμε ότι αν βρεθεί κάποιο ζευγάρι άνισων τιμών καταλήγ

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αλγόριθμος Συμμετρικός\_Πίνακας

Δεδομένα //  $N, A$

συμμετρικός ← Αληθής ! Έστω πίνακας συμμετρικός

Για  $i$  από 1 μέχρι  $(N \text{ div } 2)$  ! Έλεγχος μέχρι το μέσο του πίνακα

Αν  $A[i] \neq A[N+1-i]$  τότε ! Αν βρεθεί έστω ένα ζευγάρι άνισων τιμών, τότε ο πίνακας δεν είναι συμμετρικός

συμμετρικός ← Ψευδής

Τέλος\_αν

Τέλος\_επανάληψης

Αν συμμετρικός = Αληθής τότε

Γράψε "Συμμετρικός πίνακας"

Αλλιώς

Γράψε "Όχι Συμμετρικός πίνακας"

Τέλος\_αν

Τέλος Συμμετρικός\_Πίνακας

Παρατηρούμε ότι αν βρεθεί κάποιο ζευγάρι άνισων τιμών καταλήγουμε στο συμπέρασμα

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας

**Δεδομένα** //  $N, A$

συμμετρικός ← Αληθής ! Έστω πίνακας συμμετρικός

**Για**  $i$  από 1 μέχρι  $(N \text{ div } 2)$  ! Έλεγχος μέχρι το μέσο του πίνακα

**Αν**  $A[i] \neq A[N+1-i]$  **τότε** ! Αν βρεθεί έστω ένα ζευγάρι άνισων τιμών ,τότε ο πίνακας δεν είναι συμμετρικός

συμμετρικός ← Ψευδής

Τέλος\_αν

Τέλος\_επανάληψης

**Αν** συμμετρικός = Αληθής **τότε**

Γράψε "Συμμετρικός πίνακας"

Αλλιώς

Γράψε "Όχι Συμμετρικός πίνακας"

Τέλος\_αν

**Τέλος** Συμμετρικός\_Πίνακας

Παρατηρούμε ότι αν βρεθεί κάποιο ζευγάρι άνισων τιμών καταλήγουμε στο συμπέρασμα αλλά ο έλεγχος συνεχίζεται.

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αλγόριθμος Συμμετρικός\_Πίνακας

Δεδομένα //  $N, A$ 

συμμετρικός ← Αληθής ! Έστω πίνακας συμμετρικός

Για  $i$  από 1 μέχρι  $(N \text{ div } 2)$  ! Έλεγχος μέχρι το μέσο του πίνακαΑν  $A[i] \neq A[N+1-i]$  τότε ! Αν βρεθεί έστω ένα ζευγάρι άνισων τιμών, τότε ο πίνακας δεν είναι συμμετρικός

συμμετρικός ← Ψευδής

Τέλος\_αν

Τέλος\_επανάληψης

Αν συμμετρικός = Αληθής τότε

Γράψε "Συμμετρικός πίνακας"

Αλλιώς

Γράψε "Όχι Συμμετρικός πίνακας"

Τέλος\_αν

Τέλος Συμμετρικός\_Πίνακας

Παρατηρούμε ότι αν βρεθεί κάποιο ζευγάρι άνισων τιμών καταλήγουμε στο συμπέρασμα αλλά ο έλεγχος συνεχίζεται.

Αυτό οφείλεται στη δομή Για,

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αλγόριθμος Συμμετρικός\_Πίνακας

Δεδομένα //  $N, A$ 

συμμετρικός ← Αληθής ! Έστω πίνακας συμμετρικός

Για  $i$  από 1 μέχρι  $(N \text{ div } 2)$  ! Έλεγχος μέχρι το μέσο του πίνακαΑν  $A[i] \neq A[N+1-i]$  τότε ! Αν βρεθεί έστω ένα ζευγάρι άνισων τιμών, τότε ο πίνακας δεν είναι συμμετρικός

συμμετρικός ← Ψευδής

Τέλος\_αν

Τέλος\_επανάληψης

Αν συμμετρικός = Αληθής τότε

Γράψε "Συμμετρικός πίνακας"

Αλλιώς

Γράψε "Όχι Συμμετρικός πίνακας"

Τέλος\_αν

Τέλος Συμμετρικός\_Πίνακας

Παρατηρούμε ότι αν βρεθεί κάποιο ζευγάρι άνισων τιμών καταλήγουμε στο συμπέρασμα αλλά ο έλεγχος συνεχίζεται.

Αυτό οφείλεται στη δομή Για, η οποία θα εκτελεστεί για όλες τις επαναλήψεις.

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας

**Δεδομένα** //  $N, A$

συμμετρικός  $\leftarrow$  Αληθής ! Έστω πίνακας συμμετρικός

**Για**  $i$  από 1 μέχρι  $(N \text{ div } 2)$  ! Έλεγχος μέχρι το μέσο του πίνακα

**Αν**  $A[i] \neq A[N+1-i]$  **τότε** ! Αν βρεθεί έστω ένα ζευγάρι άνισων τιμών, τότε ο πίνακας δεν είναι συμμετρικός

συμμετρικός  $\leftarrow$  Ψευδής

Τέλος\_αν

Τέλος\_επανάληψης

**Αν** συμμετρικός = Αληθής **τότε**

Γράψε "Συμμετρικός πίνακας"

Αλλιώς

Γράψε "Όχι Συμμετρικός πίνακας"

Τέλος\_αν

**Τέλος** Συμμετρικός\_Πίνακας

Παρατηρούμε ότι αν βρεθεί κάποιο ζευγάρι άνισων τιμών καταλήγουμε στο συμπέρασμα αλλά ο έλεγχος συνεχίζεται.

Αυτό οφείλεται στη δομή Για, η οποία θα εκτελεστεί για όλες τις επαναλήψεις.

Αυτή η ατέλεια διορθώνεται

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

## Πίνακες

1.20

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας

**Δεδομένα** //  $N, A$

συμμετρικός ← Αληθής ! Έστω πίνακας συμμετρικός

**Για**  $i$  από 1 μέχρι  $(N \text{ div } 2)$  ! Έλεγχος μέχρι το μέσο του πίνακα

**Αν**  $A[i] \neq A[N+1-i]$  **τότε** ! Αν βρεθεί έστω ένα ζευγάρι άνισων τιμών ,τότε ο πίνακας δεν είναι συμμετρικός

συμμετρικός ← Ψευδής

Τέλος\_αν

Τέλος\_επανάληψης

**Αν** συμμετρικός = Αληθής **τότε**

Γράψε "Συμμετρικός πίνακας"

Αλλιώς

Γράψε "Όχι Συμμετρικός πίνακας"

Τέλος\_αν

Τέλος Συμμετρικός\_Πίνακας

Παρατηρούμε ότι αν βρεθεί κάποιο ζευγάρι άνισων τιμών καταλήγουμε στο συμπέρασμα αλλά ο έλεγχος συνεχίζεται.

Αυτό οφείλεται στη δομή Για, η οποία θα εκτελεστεί για όλες τις επαναλήψεις.

Αυτή η ατέλεια διορθώνεται με τον παρακάτω αλγόριθμο

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

## Πίνακες

1.20

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας

**Δεδομένα** //  $N, A$

συμμετρικός ← Αληθής ! Έστω πίνακας συμμετρικός

**Για**  $i$  από 1 μέχρι  $(N \text{ div } 2)$  ! Έλεγχος μέχρι το μέσο του πίνακα

**Αν**  $A[i] \neq A[N+1-i]$  **τότε** ! Αν βρεθεί έστω ένα ζευγάρι άνισων τιμών ,τότε ο πίνακας δεν είναι συμμετρικός

συμμετρικός ← Ψευδής

Τέλος\_αν

Τέλος\_επανάληψης

**Αν** συμμετρικός = Αληθής **τότε**

Γράψε "Συμμετρικός πίνακας"

Αλλιώς

Γράψε "Όχι Συμμετρικός πίνακας"

Τέλος\_αν

Τέλος Συμμετρικός\_Πίνακας

Παρατηρούμε ότι αν βρεθεί κάποιο ζευγάρι άνισων τιμών καταλήγουμε στο συμπέρασμα αλλά ο έλεγχος συνεχίζεται.

Αυτό οφείλεται στη δομή Για, η οποία θα εκτελεστεί για όλες τις επαναλήψεις.

Αυτή η ατέλεια διορθώνεται με τον παρακάτω αλγόριθμο

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.



## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αλγόριθμος Συμμετρικός\_Πίνακας

Δεδομένα //  $N, A//$ 

Τέλος Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας

**Δεδομένα** //  $N, A$

συμμετρικός ← Αληθής ! Έστω πίνακας συμμετρικός

**Τέλος** Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας

**Δεδομένα** //  $N, A$

συμμετρικός  $\leftarrow$  Αληθής ! Έστω πίνακας συμμετρικός

$i \leftarrow 1$

Όσο

**Τέλος** Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας**Δεδομένα** //  $N, A$ συμμετρικός  $\leftarrow$  Αληθής ! Έστω πίνακας συμμετρικός $i \leftarrow 1$ **Όσο** ( $i \leq N \text{ div } 2$ ) **και** (συμμετρικός = Αληθής)**Τέλος** Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας**Δεδομένα** //  $N, A$ συμμετρικός  $\leftarrow$  Αληθής ! Έστω πίνακας συμμετρικός $i \leftarrow 1$ **Όσο** ( $i \leq N \text{ div } 2$ ) **και** (συμμετρικός = Αληθής)**Τέλος** Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας**Δεδομένα** //  $N, A$  //συμμετρικός  $\leftarrow$  Αληθής ! Έστω πίνακας συμμετρικός $i \leftarrow 1$ **Όσο** ( $i \leq N \text{ div } 2$ ) και (συμμετρικός = Αληθής) **επανάλαβε****Αν**  $A[i] \neq A[N+1-i]$ **Τέλος** Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας**Δεδομένα** //  $N, A$  //συμμετρικός  $\leftarrow$  Αληθής ! Έστω πίνακας συμμετρικός $i \leftarrow 1$ **Όσο** ( $i \leq N \text{ div } 2$ ) και (συμμετρικός = Αληθής) **επανάλαβε****Αν**  $A[i] \neq A[N+1-i]$  **τότε****Τέλος** Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας**Δεδομένα** //  $N, A$  //συμμετρικός  $\leftarrow$  Αληθής ! Έστω πίνακας συμμετρικός $i \leftarrow 1$ **Όσο** ( $i \leq N \text{ div } 2$ ) και (συμμετρικός = Αληθής) **επανάλαβε****Αν**  $A[i] \neq A[N+1-i]$  **τότε****Τέλος** Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας**Δεδομένα** //  $N, A$  //συμμετρικός  $\leftarrow$  Αληθής ! Έστω πίνακας συμμετρικός $i \leftarrow 1$ **Όσο** ( $i \leq N \text{ div } 2$ ) και (συμμετρικός = Αληθής) **επανάλαβε****Αν**  $A[i] \neq A[N+1-i]$  **τότε**συμμετρικός  $\leftarrow$  Ψευδής**Τέλος** Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας**Δεδομένα** //  $N, A$  //συμμετρικός  $\leftarrow$  Αληθής ! Έστω πίνακας συμμετρικός $i \leftarrow 1$ Όσο ( $i \leq N \text{ div } 2$ ) και (συμμετρικός = Αληθής) επανάλαβεΑν  $A[i] \neq A[N+1-i]$  τότεσυμμετρικός  $\leftarrow$  Ψευδής**Τέλος** Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας**Δεδομένα** //  $N, A$  //συμμετρικός  $\leftarrow$  Αληθής ! Έστω πίνακας συμμετρικός $i \leftarrow 1$ Όσο ( $i \leq N \text{ div } 2$  και (συμμετρικός = Αληθής)) επανάλαβεΑν  $A[i] \neq A[N+1-i]$  τότεσυμμετρικός  $\leftarrow$  Ψευδής**Τέλος** Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας**Δεδομένα** //  $N, A$  //συμμετρικός  $\leftarrow$  Αληθής ! Έστω πίνακας συμμετρικός $i \leftarrow 1$ **Όσο** ( $i \leq N \text{ div } 2$ ) και (συμμετρικός = Αληθής) **επανάλαβε****Αν**  $A[i] \neq A[N+1-i]$  **τότε** ! Αν είναι τότεσυμμετρικός  $\leftarrow$  Ψευδής**Τέλος** Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας**Δεδομένα** //  $N, A$  //συμμετρικός  $\leftarrow$  Αληθής ! Έστω πίνακας συμμετρικός $i \leftarrow 1$ **Όσο** ( $i \leq N \text{ div } 2$ ) και (συμμετρικός = Αληθής) **επανάλαβε****Αν**  $A[i] \neq A[N+1-i]$  **τότε** ! Αν είναι τότε βγές έξω από την επανάληψη  
συμμετρικός  $\leftarrow$  Ψευδής**Τέλος** Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας

**Δεδομένα** //  $N, A$  //

συμμετρικός  $\leftarrow$  Αληθής ! Έστω πίνακας συμμετρικός

$i \leftarrow 1$

**Όσο** ( $i \leq N \text{ div } 2$ ) και (συμμετρικός = Αληθής) **επανάλαβε**

**Αν**  $A[i] \neq A[N+1-i]$  **τότε**

! Αν είναι τότε βγές έξω από την επανάληψη

συμμετρικός  $\leftarrow$  Ψευδής

**Αλλιώς\_αν**

! Αν δεν είναι τότε έλεγξε το επόμενο

**Τέλος** Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας**Δεδομένα** //  $N, A$  //συμμετρικός  $\leftarrow$  Αληθής ! Έστω πίνακας συμμετρικός $i \leftarrow 1$ **Όσο** ( $i \leq N \text{ div } 2$ ) και (συμμετρικός = Αληθής) **επανάλαβε****Αν**  $A[i] \neq A[N+1-i]$  **τότε**

! Αν είναι τότε βγές έξω από την επανάληψη

συμμετρικός  $\leftarrow$  Ψευδής**Αλλιώς\_αν**

! Αν δεν είναι τότε έλεγξε το επόμενο

 $i \leftarrow i+1$ **Τέλος** Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας

**Δεδομένα** //  $N, A$  //

συμμετρικός  $\leftarrow$  Αληθής ! Έστω πίνακας συμμετρικός

$i \leftarrow 1$

**Όσο** ( $i \leq N \text{ div } 2$ ) και (συμμετρικός = Αληθής) **επανάλαβε**

**Αν**  $A[i] \neq A[N+1-i]$  **τότε** ! Αν είναι τότε βγές έξω από την επανάληψη

συμμετρικός  $\leftarrow$  Ψευδής

**Αλλιώς\_αν** ! Αν δεν είναι τότε έλεγξε το επόμενο

$i \leftarrow i+1$  ! Εντ.

**Τέλος** Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας

**Δεδομένα** //  $N, A$  //

συμμετρικός  $\leftarrow$  Αληθής ! Έστω πίνακας συμμετρικός

$i \leftarrow 1$

**Όσο** ( $i \leq N \text{ div } 2$ ) και (συμμετρικός = Αληθής) **επανάλαβε**

**Αν**  $A[i] \neq A[N+1-i]$  **τότε** ! Αν είναι τότε βγές έξω από την επανάληψη

συμμετρικός  $\leftarrow$  Ψευδής

**Αλλιώς\_αν** ! Αν δεν είναι τότε έλεγξε το επόμενο

$i \leftarrow i+1$  ! Εντολή Αλλαγής

**Τέλος\_αν**

**Τέλος\_επανα**

**Τέλος** Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας

**Δεδομένα** //  $N, A$  //

συμμετρικός  $\leftarrow$  Αληθής ! Έστω πίνακας συμμετρικός

$i \leftarrow 1$

**Όσο** ( $i \leq N \text{ div } 2$ ) και (συμμετρικός = Αληθής) **επανάλαβε**

**Αν**  $A[i] \neq A[N+1-i]$  **τότε** ! Αν είναι τότε βγές έξω από την επανάληψη

συμμετρικός  $\leftarrow$  Ψευδής

**Αλλιώς\_αν** ! Αν δεν είναι τότε έλεγξε το επόμενο

$i \leftarrow i+1$  ! Εντολή Αλλαγής

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Αν** συμμετρικός = Αληθής **τότε**

**Τέλος** Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας

**Δεδομένα** //  $N, A$

συμμετρικός  $\leftarrow$  Αληθής ! Έστω πίνακας συμμετρικός

$i \leftarrow 1$

**Όσο** ( $i \leq N \text{ div } 2$ ) και (συμμετρικός = Αληθής) **επανάλαβε**

**Αν**  $A[i] \neq A[N+1-i]$  **τότε** ! Αν είναι τότε βγές έξω από την επανάληψη

συμμετρικός  $\leftarrow$  Ψευδής

**Αλλιώς\_αν** ! Αν δεν είναι τότε έλεγξε το επόμενο

$i \leftarrow i+1$  ! Εντολή Αλλαγής

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Αν** συμμετρικός = Αληθής **τότε**

**Γράψε** "Συμμετρικός πίνακας"

**Τέλος** Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας

**Δεδομένα** //  $N, A$

συμμετρικός  $\leftarrow$  Αληθής ! Έστω πίνακας συμμετρικός

$i \leftarrow 1$

Όσο ( $i \leq N \text{ div } 2$ ) και (συμμετρικός = Αληθής) επανάλαβε

Αν  $A[i] \neq A[N+1-i]$  τότε ! Αν είναι τότε βγές έξω από την επανάληψη

συμμετρικός  $\leftarrow$  Ψευδής

Αλλιώς\_αν ! Αν δεν είναι τότε έλεγξε το επόμενο

$i \leftarrow i+1$  ! Εντολή Αλλαγής

Τέλος\_αν

Τέλος\_επανάληψης

Αν συμμετρικός = Αληθής τότε

Γράψε "Συμμετρικός πίνακας"

Αλλιώς

Γράψε "Όχι Συμμετρικός πίνακας"

**Τέλος** Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας

**Δεδομένα** //  $N, A$

συμμετρικός  $\leftarrow$  Αληθής ! Έστω πίνακας συμμετρικός

$i \leftarrow 1$

**Όσο** ( $i \leq N \text{ div } 2$ ) και (συμμετρικός = Αληθής) **επανάλαβε**

**Αν**  $A[i] \neq A[N+1-i]$  **τότε** ! Αν είναι τότε βγές έξω από την επανάληψη

συμμετρικός  $\leftarrow$  Ψευδής

**Αλλιώς\_αν** ! Αν δεν είναι τότε έλεγξε το επόμενο

$i \leftarrow i+1$  ! Εντολή Αλλαγής

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Αν** συμμετρικός = Αληθής **τότε**

**Γράψε** "Συμμετρικός πίνακας"

**Αλλιώς**

**Γράψε** "Όχι Συμμετρικός πίνακας"

**Τέλος\_αν**

**Τέλος** Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

**Αλγόριθμος** Συμμετρικός\_Πίνακας

**Δεδομένα** //  $N, A$

συμμετρικός  $\leftarrow$  Αληθής ! Έστω πίνακας συμμετρικός

$i \leftarrow 1$

**Όσο** ( $i \leq N \text{ div } 2$ ) και (συμμετρικός = Αληθής) **επανάλαβε**

**Αν**  $A[i] \neq A[N+1-i]$  **τότε** ! Αν είναι τότε βγές έξω από την επανάληψη

συμμετρικός  $\leftarrow$  Ψευδής

**Αλλιώς\_αν** ! Αν δεν είναι τότε έλεγξε το επόμενο

$i \leftarrow i+1$  ! Εντολή Αλλαγής

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Αν** συμμετρικός = Αληθής **τότε**

Γράψε "Συμμετρικός πίνακας"

**Αλλιώς**

Γράψε "Όχι Συμμετρικός πίνακας"

**Τέλος\_αν**

**Τέλος** Συμμετρικός\_Πίνακας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αλγόριθμος Συμμετρικός\_Πίνακας

Δεδομένα //  $N, A$ 1-1  $\text{συμμετρικός} \leftarrow \text{Αληθής}$  ! Έστω πίνακας συμμετρικός $i \leftarrow 1$ **Όσο** ( $i \leq N \text{ div } 2$ ) και ( $\text{συμμετρικός} = \text{Αληθής}$ ) **επανάλαβε****Αν**  $A[i] \neq A[N+1-i]$  **τότε** ! Αν είναι τότε βγές έξω από την επανάληψη $\text{συμμετρικός} \leftarrow \text{Ψευδής}$ **Αλλιώς\_αν** ! Αν δεν είναι τότε έλεγξε το επόμενο $i \leftarrow i+1$  ! Εντολή Αλλαγής**Τέλος\_αν****Τέλος\_επανάληψης****Αν**  $\text{συμμετρικός} = \text{Αληθής}$  **τότε**

Γράψε "Συμμετρικός πίνακας"

**Αλλιώς**

Γράψε "Όχι Συμμετρικός πίνακας"

**Τέλος\_αν****Τέλος** Συμμετρικός\_Πίνακας

Προσοχή

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αλγόριθμος Συμμετρικός\_Πίνακας

Δεδομένα //  $N, A$ συμμετρικός  $\leftarrow$  Αληθής ! Έστω πίνακας συμμετρικός $i \leftarrow 1$ Όσο ( $i \leq N \text{ div } 2$ ) και (συμμετρικός = Αληθής) επανάλαβεΑν  $A[i] \neq A[N+1-i]$  τότε ! Αν είναι τότε βγές έξω από την επανάληψησυμμετρικός  $\leftarrow$  Ψευδής

Αλλιώς\_αν ! Αν δεν είναι τότε έλεγξε το επόμενο

 $i \leftarrow i+1$  ! Εντολή Αλλαγής

Τέλος\_αν

Τέλος\_επανάληψης

Αν συμμετρικός = Αληθής τότε

Γράψε "Συμμετρικός πίνακας"

Αλλιώς

Γράψε "Όχι Συμμετρικός πίνακας"

Τέλος\_αν

Τέλος Συμμετρικός\_Πίνακας

Προσοχή

 $i - N + 1 - i$

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

Αλγόριθμος Συμμετρικός\_Πίνακας

Δεδομένα //  $N, A$ 

συμμετρικός ← Αληθής ! Έστω πίνακας συμμετρικός

 $i \leftarrow 1$ Όσο ( $i \leq N \text{ div } 2$ ) και (συμμετρικός = Αληθής) επανάλαβεΑν  $A[i] \neq A[N+1-i]$  τότε ! Αν είναι τότε βγές έξω από την επανάληψη

συμμετρικός ← Ψευδής

Αλλιώς\_αν ! Αν δεν είναι τότε έλεγξε το επόμενο

 $i \leftarrow i+1$  ! Εντολή Αλλαγής

Τέλος\_αν

Τέλος\_επανάληψης

Αν συμμετρικός = Αληθής τότε

Γράψε "Συμμετρικός πίνακας"

Αλλιώς

Γράψε "Όχι Συμμετρικός πίνακας"

Τέλος\_αν

Τέλος Συμμετρικός\_Πίνακας

Προσοχή

 $i \leftarrow N+1-i$ Συμμετρία στον  
μονοδιάστατο πίνακα

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.20

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα ελέγχει αν τα συμμετρικά του στοιχεία είναι ίσα.

### Αλγόριθμος Συμμετρικός\_Πίνακας

Δεδομένα //  $N, A$

συμμετρικός  $\leftarrow$  Αληθής ! Έστω πίνακας συμμετρικός  
 $i \leftarrow 1$

Όσο ( $i \leq N \text{ div } 2$ ) και (συμμετρικός = Αληθής) επανάλαβε

Αν  $A[i] \neq A[N+1-i]$  τότε ! Αν είναι τότε βγές έξω από την επανάληψη  
 συμμετρικός  $\leftarrow$  Ψευδής

Αλλιώς\_αν ! Αν δεν είναι τότε έλεγξε το επόμενο  
 $i \leftarrow i+1$  ! Εντολή Αλλαγής

Τέλος\_αν

Τέλος\_επανάληψης

Αν συμμετρικός = Αληθής τότε  
 Γράψε "Συμμετρικός πίνακας"

Αλλιώς  
 Γράψε "Όχι Συμμετρικός πίνακας"

Τέλος\_αν

Τέλος Συμμετρικός\_Πίνακας

Προσοχή

$i \leftarrow N+1-i$

Συμμετρία στον μονοδιάστατο πίνακα

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο  $i$  πίνακα.

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

Αλγόριθμος Εύρεση\_ελαχίστου

Τέλος Εύρεση\_ελαχίστου

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Εύρεση\_ελαχίστου

**Δεδομένα** //  $N, A$

ελάχιστος  $\leftarrow A[1]$

**Τέλος** Εύρεση\_ελαχίστου

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Εύρεση\_ελαχίστου

**Δεδομένα** //  $N, A$

ελάχιστος  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

**Τέλος** Εύρεση\_ελαχίστου

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Εύρεση\_ελαχίστου

**Δεδομένα** //  $N, A$

ελάχιστος  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

**Για**

**Τέλος** Εύρεση\_ελαχίστου

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Εύρεση\_ελαχίστου

**Δεδομένα** //  $N, A$

**ελάχιστος**  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

**Για**  $i$  από 2 μέχρι  $N$

**Αν**

**Τέλος** Εύρεση\_ελαχίστου

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Εύρεση\_ελαχίστου

**Δεδομένα** //  $N, A$

**ελάχιστος**  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

**Για**  $i$  από 2 μέχρι  $N$

**Αν**  $A[i] < \text{ελάχιστος}$  τότε

**Τέλος** Εύρεση\_ελαχίστου

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

Αλγόριθμος Εύρεση\_ελαχίστου

Δεδομένα //  $N, A$

ελάχιστος  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

Για  $i$  από 2 μέχρι  $N$

Αν  $A[i] < \text{ελάχιστος}$  τότε

    ελάχιστος  $\leftarrow A[i]$

Τέλος\_αν

Τέλος\_επανάλη

Τέλος Εύρεση\_ελαχίστου

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

## Πίνακες

1.23

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Εύρεση\_ελαχίστου

**Δεδομένα** //  $N, A$

**ελάχιστος**  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

**Για**  $i$  από 2 μέχρι  $N$

**Αν**  $A[i] < \text{ελάχιστος}$  τότε

**ελάχιστος**  $\leftarrow A[i]$

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Γράψε** "Το ελάχιστο είναι ο αριθμός", ελάχιστο

**Τέλος** Εύρεση\_ελαχίστου

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

## Πίνακες

1.23

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

```
Αλγόριθμος  Εύρεση_ελαχίστου
  Δεδομένα  // N, A//
  ελάχιστος ← A[1] ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο
  Για  i από 2 μέχρι N
    Αν  A[i] < ελάχιστος τότε
      ελάχιστος ← A[i]
  Τέλος_αν
  Τέλος_επανάληψης
  Γράψε  "Το ελάχιστο είναι ο αριθμός", ελάχιστο
Τέλος  Εύρεση_ελαχίστου
```



## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

## Πίνακες

1.23

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Εύρεση\_ελαχίστου

**Δεδομένα** //  $N, A$

**ελάχιστος**  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

**Για**  $i$  από 2 μέχρι  $N$

**Αν**  $A[i] < \text{ελάχιστος}$  τότε

**ελάχιστος**  $\leftarrow A[i]$

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Γράψε** "Το ελάχιστο είναι ο αριθμός", ελάχιστο

**Τέλος** Εύρεση\_ελαχίστου

Αυτός ο αλγόριθμος δεν μπορεί να

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

Αλγόριθμος Εύρεση\_ελαχίστου

Δεδομένα //  $N, A$  //

ελάχιστος  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

Για  $i$  από 2 μέχρι  $N$

Αν  $A[i] < \text{ελάχιστος}$  τότε

ελάχιστος  $\leftarrow A[i]$

Τέλος\_αν

Τέλος\_επανάληψης

Γράψε "Το ελάχιστο είναι ο αριθμός", ελάχιστο

Τέλος Εύρεση\_ελαχίστου

Αυτός ο αλγόριθμος δεν μπορεί να εκτυπώσει σε ποια θέση του πίνακα βρίσκεται αυτό.

Είν

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος  Εύρεση_ελαχίστου
  Δεδομένα  // N, A//
  ελάχιστος ← A[1] ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο
  Για  i από 2 μέχρι N
    Αν  A[i] < ελάχιστος τότε
      ελάχιστος ← A[i]
    Τέλος_αν
  Τέλος_επανάληψης
  Γράψε  "Το ελάχιστο είναι ο αριθμός", ελάχιστο
Τέλος  Εύρεση_ελαχίστου
  
```

Αυτός ο αλγόριθμος δεν μπορεί να εκτυπώσει σε ποια θέση του πίνακα βρίσκεται αυτό.  
Είναι λάθος να θεωρείται η θέση  $i$ , η θέση του ελαχίστου,

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

Αλγόριθμος Εύρεση\_ελαχίστου

Δεδομένα //  $N, A$

ελάχιστος  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

Για  $i$  από 2 μέχρι  $N$

Αν  $A[i] < \text{ελάχιστος}$  τότε

ελάχιστος  $\leftarrow A[i]$

Τέλος\_αν

Τέλος\_επανάληψης

Γράψε "Το ελάχιστο είναι ο αριθμός", ελάχιστο

Τέλος Εύρεση\_ελαχίστου

Αυτός ο αλγόριθμος δεν μπορεί να εκτυπώσει σε ποια θέση του πίνακα βρίσκεται αυτό.  
Είναι λάθος να θεωρείται η θέση  $i$ , η θέση του ελαχίστου, διότι το  $i$  έχει τιμή ίση με την τιμή

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος  Εύρεση_ελαχίστου
  Δεδομένα  // N, A//
  ελάχιστος ← A[1] ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο
  Για  i από 2 μέχρι N
    Αν  A[i] < ελάχιστος τότε
      ελάχιστος ← A[i]
  Τέλος_αν
  Τέλος_επανάληψης
  Γράψε  "Το ελάχιστο είναι ο αριθμός", ελάχιστο
Τέλος  Εύρεση_ελαχίστου
  
```

Αυτός ο αλγόριθμος δεν μπορεί να εκτυπώσει σε ποια θέση του πίνακα βρίσκεται αυτό.  
Είναι λάθος να θεωρείται η θέση  $i$ , η θέση του ελαχίστου,  
διότι το  $i$  έχει τιμή ίση με την τιμή που βγαίνει από την επανάληψη δηλαδή  $N$ .

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος  Εύρεση_ελαχίστου
  Δεδομένα  // N, A//
  ελάχιστος ← A[1] ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο
  Για  i από 2 μέχρι N
    Αν  A[i] < ελάχιστος τότε
      ελάχιστος ← A[i]
  Τέλος_αν
  Τέλος_επανάληψης
  Γράψε  "Το ελάχιστο είναι ο αριθμός", ελάχιστο
Τέλος  Εύρεση_ελαχίστου
  
```

Αυτός ο αλγόριθμος δεν μπορεί να εκτυπώσει σε ποια θέση του πίνακα βρίσκεται αυτό.  
Είναι λάθος να θεωρείται η θέση  $i$ , η θέση του ελαχίστου, διότι το  $i$  έχει τιμή ίση με την τιμή που βγαίνει από την επανάληψη δηλαδή  $N$ .  
Επομένως απαιτείται η χρήση μιας

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Εύρεση\_ελαχίστου

**Δεδομένα** //  $N, A$

**ελάχιστος**  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

**Για**  $i$  από 2 μέχρι  $N$

**Αν**  $A[i] < \text{ελάχιστος}$  τότε

**ελάχιστος**  $\leftarrow A[i]$

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Γράψε** "Το ελάχιστο είναι ο αριθμός", ελάχιστο

**Τέλος** Εύρεση\_ελαχίστου

Αυτός ο αλγόριθμος δεν μπορεί να εκτυπώσει σε ποια θέση του πίνακα βρίσκεται αυτό.

Είναι λάθος να θεωρείται η θέση  $i$ , η θέση του ελαχίστου, διότι το  $i$  έχει τιμή ίση με την τιμή που βγαίνει από την επανάληψη δηλαδή  $N$ .

Επομένως απαιτείται η χρήση μιας επιπλέον μεταβλητής, η οποία θα αποθηκεύει τη θέση του ελαχίστου

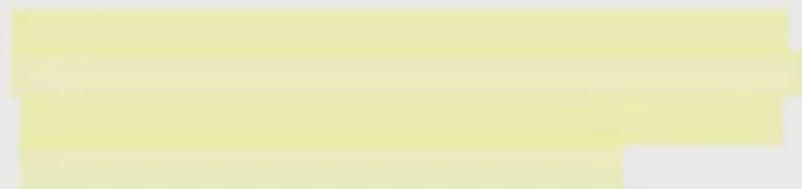
## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.



## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

Αλ

Εύρ

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

Αλγόριθμος Εύρεση\_ελαχίστου

Δεδομέν

Τέλος Εύρεση\_ελαχίστου

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Εύρεση\_ελαχίστου

**Δεδομένα** //  $N, A$

ελάχιστος  $\leftarrow A[1]$

**Τέλος** Εύρεση\_ελαχίστου

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Εύρεση\_ελαχίστου

**Δεδομένα** //  $N, A$

ελάχιστος  $\leftarrow A[1]$

θέση\_ελαχίστου  $\leftarrow 1$

**Τέλος** Εύρεση\_ελαχίστου

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Εύρεση\_ελαχίστου

**Δεδομένα** //  $N, A$

ελάχιστος  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

θέση\_ελαχίστου  $\leftarrow 1$

**Για**

**Τέλος** Εύρεση\_ελαχίστου

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Εύρεση\_ελαχίστου

**Δεδομένα** //  $N, A$

ελάχιστος  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

θέση\_ελαχίστου  $\leftarrow 1$

**Για**  $i$  από 2 μέχρι  $N$

**Τέλος** Εύρεση\_ελαχίστου

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Εύρεση\_ελαχίστου

**Δεδομένα** //  $N, A$

ελάχιστος  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

θέση\_ελαχίστου  $\leftarrow 1$

**Για**  $i$  από 2 μέχρι  $N$

**Αν**  $A[i] <$

**Τέλος** Εύρεση\_ελαχίστου

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Εύρεση\_ελαχίστου

**Δεδομένα** //  $N, A$

ελάχιστος  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

θέση\_ελαχίστου  $\leftarrow 1$

**Για**  $i$  από 2 μέχρι  $N$

**Αν**  $A[i] <$  ελάχιστος **τότε**

ελάχιστος  $\leftarrow A[i]$

θέση\_

**Τέλος** Εύρεση\_ελαχίστου

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Εύρεση\_ελαχίστου

**Δεδομένα** //  $N, A$

ελάχιστος  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

θέση\_ελαχίστου  $\leftarrow 1$

**Για**  $i$  από 2 μέχρι  $N$

**Αν**  $A[i] <$  ελάχιστος **τότε**

ελάχιστος  $\leftarrow A[i]$

θέση\_ελαχίστου  $\leftarrow i$

**Τέλος\_αν**

**Τέλος\_επανάληψη**

**Τέλος** Εύρεση\_ελαχίστου

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Εύρεση\_ελαχίστου

**Δεδομένα** //  $N, A$

ελάχιστος  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

θέση\_ελαχίστου  $\leftarrow 1$

**Για**  $i$  από 2 μέχρι  $N$

**Αν**  $A[i] <$  ελάχιστος **τότε**

ελάχιστος  $\leftarrow A[i]$

θέση\_ελαχίστου  $\leftarrow i$

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Γράψε**

**Τέλος** Εύρεση\_ελαχίστου

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Εύρεση\_ελαχίστου

**Δεδομένα** //  $N, A$

ελάχιστος  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

θέση\_ελαχίστου  $\leftarrow 1$

**Για**  $i$  από 2 μέχρι  $N$

**Αν**  $A[i] <$  ελάχιστος **τότε**

ελάχιστος  $\leftarrow A[i]$

θέση\_ελαχίστου  $\leftarrow i$

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Γράψε** "Το ελάχιστο είναι ο αριθμός", ελάχιστος

**Τέλος** Εύρεση\_ελαχίστου

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Εύρεση\_ελαχίστου

**Δεδομένα** //  $N, A$

ελάχιστος  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο  
θέση\_ελαχίστου  $\leftarrow 1$

**Για**  $i$  από 2 μέχρι  $N$

**Αν**  $A[i] <$  ελάχιστος **τότε**

ελάχιστος  $\leftarrow A[i]$

θέση\_ελαχίστου  $\leftarrow i$

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Γράψε** "Το ελάχιστο είναι ο αριθμός", ελάχιστος

**Γράψε** "Η θέση του ελαχίστου είναι", θέση\_ελαχίστου

**Τέλος** Εύρεση\_ελαχίστου

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Εύρεση\_ελαχίστου

**Δεδομένα** //  $N, A$

ελάχιστος  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

θέση\_ελαχίστου  $\leftarrow 1$

**Για**  $i$  από 2 μέχρι  $N$

**Αν**  $A[i] <$  ελάχιστος **τότε**

ελάχιστος  $\leftarrow A[i]$

θέση\_ελαχίστου  $\leftarrow i$

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Γράψε** "Το ελάχιστο είναι ο αριθμός", ελάχιστος

**Γράψε** "Η θέση του ελαχίστου είναι", θέση\_ελαχίστου

**Τέλος** Εύρεση\_ελαχίστου

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Εύρεση\_ελαχίστου

**Δεδομένα** //  $N, A$

ελάχιστος  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

θέση\_ελαχίστου  $\leftarrow 1$

**Για**  $i$  από 2 μέχρι  $N$

**Αν**  $A[i] <$  ελάχιστος **τότε**

ελάχιστος  $\leftarrow A[i]$

θέση\_ελαχίστου  $\leftarrow i$

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Γράψε** "Το ελάχιστο είναι ο αριθμός", ελάχιστος

**Γράψε** "Η θέση του ελαχίστου είναι", θέση\_ελαχίστου

**Τέλος** Εύρεση\_ελαχίστου

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος  Εύρεση_ελαχίστου
  Δεδομένα  // N, A//
  ελάχιστος ← A[1]  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο
  θέση_ελαχίστου ← 1

  Για  i από 2 μέχρι N

    Αν A[i] < ελάχιστος τότε
      ελάχιστος ← A[i]
      θέση_ελαχίστου ← i
    Τέλος_αν
  Τέλος_επανάληψης

  Γράψε  "Το ελάχιστο είναι ο αριθμός", ελάχιστος
  Γράψε  "Η θέση του ελαχίστου είναι", θέση_ελαχίστου
Τέλος  Εύρεση_ελαχίστου
  
```

Τι γίνεται όμως στην περίπτωση που έχουμε παραπάνω από ένα

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος  Εύρεση_ελαχίστου
  Δεδομένα // N, A//
  ελάχιστος ← A[1]  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο
  θέση_ελαχίστου ← 1

  Για i από 2 μέχρι N
    Αν A[i] < ελάχιστος τότε
      ελάχιστος ← A[i]
      θέση_ελαχίστου ← i
    Τέλος_αν
  Τέλος_επανάληψης
  Γράψε "Το ελάχιστο είναι ο αριθμός", ελάχιστος
  Γράψε "Η θέση του ελαχίστου είναι", θέση_ελαχίστου
  Τέλος  Εύρεση_ελαχίστου
  
```

Τι γίνεται όμως στην περίπτωση που έχουμε παραπάνω από ένα ελάχιστα;

[ 12,13,5,-7,12,1;

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος  Εύρεση_ελαχίστου
  Δεδομένα // N, A//
  ελάχιστος ← A[1]  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο
  θέση_ελαχίστου ← 1

  Για i από 2 μέχρι N
    Αν A[i] < ελάχιστος τότε
      ελάχιστος ← A[i]
      θέση_ελαχίστου ← i
    Τέλος_αν
  Τέλος_επανάληψης
  Γράψε "Το ελάχιστο είναι ο αριθμός", ελάχιστος
  Γράψε "Η θέση του ελαχίστου είναι", θέση_ελαχίστου
Τέλος  Εύρεση_ελαχίστου
  
```

Τι γίνεται όμως στην περίπτωση που έχουμε παραπάνω από ένα ελάχιστα;

[12,13,5,-7,12,13,18,0,-7,-5,-2,4,54,]

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Εύρεση\_ελαχίστου

**Δεδομένα** //  $N, A$

ελάχιστος  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

θέση\_ελαχίστου  $\leftarrow 1$

**Για**  $i$  από 2 μέχρι  $N$

**Αν**  $A[i] <$  ελάχιστος **τότε**

ελάχιστος  $\leftarrow A[i]$

θέση\_ελαχίστου  $\leftarrow i$

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Γράψε** "Το ελάχιστο είναι ο αριθμός", ελάχιστος

**Γράψε** "Η θέση του ελαχίστου είναι", θέση\_ελαχίστου

**Τέλος** Εύρεση\_ελαχίστου

Τι γίνεται όμως στην περίπτωση που έχουμε παραπάνω από ένα ελάχιστα;

[12,13,5,-7,12,13,18,0,-7,-5,-2,4,54,]

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Εύρεση\_ελαχίστου

**Δεδομένα** //  $N, A$

ελάχιστος  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

θέση\_ελαχίστου  $\leftarrow 1$

**Για**  $i$  από 2 μέχρι  $N$

**Αν**  $A[i] <$  ελάχιστος **τότε**

ελάχιστος  $\leftarrow A[i]$

θέση\_ελαχίστου  $\leftarrow i$

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Γράψε** "Το ελάχιστο είναι ο αριθμός", ελάχιστος

**Γράψε** "Η θέση του ελαχίστου είναι", θέση\_ελαχίστου

**Τέλος** Εύρεση\_ελαχίστου

Τι γίνεται όμως στην περίπτωση που έχουμε παραπάνω από ένα ελάχιστα;

[12,13,5,-7,12,13,18,0,-7,-5,-2,4,54,]

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

### Πίνακες

1.23

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος  Εύρεση_ελαχίστου
  Δεδομένα  // N, A//
  ελάχιστος ← A[1]  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο
  θέση_ελαχίστου ← 1

  Για  i από 2 μέχρι N
    Αν A[i] < ελάχιστος τότε
      ελάχιστος ← A[i]
      θέση_ελαχίστου ← i
    Τέλος_αν
  Τέλος_επανάληψης

  Γράψε  "Το ελάχιστο είναι ο αριθμός", ελάχιστος
  Γράψε  "Η θέση του ελαχίστου είναι", θέση_ελαχίστου
Τέλος  Εύρεση_ελαχίστου
  
```

Τι γίνεται όμως στην περίπτωση που έχουμε παραπάνω από ένα ελάχιστα;

[ 12,13,5,-7, 12,13,18,0,-7,-5,-2,4,54, ]

Με αυτόν τον αλγόρ

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

### Πίνακες

1.23

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος  Εύρεση_ελαχίστου
Δεδομένα // N, A//
ελάχιστος ← A[1]  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο
θέση_ελαχίστου ← 1

Για i από 2 μέχρι N
    Αν A[i] < ελάχιστος τότε
        ελάχιστος ← A[i]
        θέση_ελαχίστου ← i
    Τέλος_αν
Τέλος_επανάληψης
Γράψε "Το ελάχιστο είναι ο αριθμός", ελάχιστος
Γράψε "Η θέση του ελαχίστου είναι", θέση_ελαχίστου
Τέλος Εύρεση_ελαχίστου
    
```

Τι γίνεται όμως στην περίπτωση που έχουμε παραπάνω από ένα ελάχιστα;

[ 12,13,5,-7, 12,13,18,0,-7,-5,-2,4,54, ]

Με αυτόν τον αλγόριθμο εκτυπώνουμε

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

### Πίνακες

1.23

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος  Εύρεση_ελαχίστου
Δεδομένα  // N, A//
ελάχιστος ← A[1]  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο
θέση_ελαχίστου ← 1

Για  i από 2 μέχρι N
    Αν  A[i] < ελάχιστος τότε
        ελάχιστος ← A[i]
        θέση_ελαχίστου ← i
    Τέλος_αν
Τέλος_επανάληψης
Γράψε  "Το ελάχιστο είναι ο αριθμός", ελάχιστος
Γράψε  "Η θέση του ελαχίστου είναι", θέση_ελαχίστου
Τέλος  Εύρεση_ελαχίστου
    
```

Τι γίνεται όμως στην περίπτωση που έχουμε παραπάνω από ένα ελάχιστα;

[ 12,13,5,-7, 12,13,18,0,-7,-5,-2,4,54, ]

Με αυτόν τον αλγόριθμο εκτυπώνουμε

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος  Εύρεση_ελαχίστου
  Δεδομένα // N, A//
  ελάχιστος ← A[1] ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο
  θέση_ελαχίστου ← 1

  Για i από 2 μέχρι N
    Αν A[i] < ελάχιστος τότε
      ελάχιστος ← A[i]
      θέση_ελαχίστου ← i
    Τέλος_αν
  Τέλος_επανάληψης
  Γράψε "Το ελάχιστο είναι ο αριθμός", ελάχιστος
  Γράψε "Η θέση του ελαχίστου είναι", θέση_ελαχίστου
Τέλος Εύρεση_ελαχίστου
  
```

Τι γίνεται όμως στην περίπτωση που έχουμε παραπάνω από ένα ελάχιστα;

[ 12,13,5, -7, 12,13,18,0, -7, -5,-2,4,54, ]

Με αυτόν τον αλγόριθμο εκτυπώνουμε

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

### Πίνακες

1.23

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος  Εύρεση_ελαχίστου
  Δεδομένα // N, A//
  ελάχιστος ← A[1] ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο
  θέση_ελαχίστου ← 1

  Για i από 2 μέχρι N
    Αν A[i] < ελάχιστος τότε
      ελάχιστος ← A[i]
      θέση_ελαχίστου ← i
    Τέλος_αν
  Τέλος_επανάληψης
  Γράψε "Το ελάχιστο είναι ο αριθμός", ελάχιστος
  Γράψε "Η θέση του ελαχίστου είναι", θέση_ελαχίστου
  Τέλος Εύρεση_ελαχίστου
  
```

Τι γίνεται όμως στην περίπτωση που έχουμε παραπάνω από ένα ελάχιστα;

[ 12,13,5,-7, 12,13,18,0,-7,-5,-2,4,54, ]

Με αυτόν τον αλγόριθμο εκτυπώνουμε

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

### Πίνακες

1.23

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος  Εύρεση_ελαχίστου
  Δεδομένα  // N, A//
  ελάχιστος ← A[1]  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο
  θέση_ελαχίστου ← 1

  Για  i από 2 μέχρι N
    Αν A[i] < ελάχιστος τότε
      ελάχιστος ← A[i]
      θέση_ελαχίστου ← i
    Τέλος_αν
  Τέλος_επανάληψης
  Γράψε  "Το ελάχιστο είναι ο αριθμός", ελάχιστος
  Γράψε  "Η θέση του ελαχίστου είναι", θέση_ελαχίστου
Τέλος  Εύρεση_ελαχίστου
  
```

Τι γίνεται όμως στην περίπτωση που έχουμε παραπάνω από ένα ελάχιστα;

[ 12,13,5,7,12,13,18,0,7,-5,-2,4,54, ]

Με αυτόν τον αλγόριθμο εκτυπώνουμε μόνο τη πρώτη θέση του ελαχίστου.

Αν επιθυμούμε να εκτυπώσουμε

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

### Πίνακες

1.23

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος  Εύρεση_ελαχίστου
Δεδομένα  // N, A//
ελάχιστος ← A[1]  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο
θέση_ελαχίστου ← 1

Για  i από 2 μέχρι N
    Αν  A[i] < ελάχιστος τότε
        ελάχιστος ← A[i]
        θέση_ελαχίστου ← i
    Τέλος_αν
Τέλος_επανάληψης
Γράψε  "Το ελάχιστο είναι ο αριθμός", ελάχιστος
Γράψε  "Η θέση του ελαχίστου είναι", θέση_ελαχίστου
Τέλος  Εύρεση_ελαχίστου
    
```

Τι γίνεται όμως στην περίπτωση που έχουμε παραπάνω από ένα ελάχιστα;

[ 12,13,5,-7, 12,13,18,0,-7,-5,-2,4,54, ]

Με αυτόν τον αλγόριθμο εκτυπώνουμε μόνο τη πρώτη θέση του ελαχίστου.

**Αν επιθυμούμε να εκτυπώσουμε το τελευταίο ελάχιστο στοιχείο.**

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

### Πίνακες

1.23

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος  Εύρεση_ελαχίστου
Δεδομένα  // N, A//
ελάχιστος ← A[1]  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο
θέση_ελαχίστου ← 1

Για  i από 2 μέχρι N
    Αν  A[i] < ελάχιστος τότε
        ελάχιστος ← A[i]
        θέση_ελαχίστου ← i
    Τέλος_αν
Τέλος_επανάληψης
Γράψε  "Το ελάχιστο είναι ο αριθμός", ελάχιστος
Γράψε  "Η θέση του ελαχίστου είναι", θέση_ελαχίστου
Τέλος  Εύρεση_ελαχίστου
    
```

Τι γίνεται όμως στην περίπτωση που έχουμε παραπάνω από ένα ελάχιστα;

[ 12,13,5,-7, 12,13,18,0,-7,-5,-2,4,54, ]

Με αυτόν τον αλγόριθμο εκτυπώνουμε μόνο τη **πρώτη** θέση του ελαχίστου.

Αν επιθυμούμε να εκτυπώσουμε το **τελευταίο** ελάχιστο στοιχείο.

Αρκεί να προστεθεί ένα = στη συνθήκη του Αν.

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

### Πίνακες

1.23

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος  Εύρεση_ελαχίστου
  Δεδομένα  // N, A//
  ελάχιστος ← A[1]  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο
  θέση_ελαχίστου ← 1

  Για  i από 2 μέχρι N
    Αν  A[i] <= ελάχιστος τότε
      ελάχιστος ← A[i]
      θέση_ελαχίστου ← i
    Τέλος_αν
  Τέλος_επανάληψης
  Γράψε  "Το ελάχιστο είναι ο αριθμός", ελάχιστος
  Γράψε  "Η θέση του ελαχίστου είναι", θέση_ελαχίστου
  Τέλος  Εύρεση_ελαχίστου
  
```

Τι γίνεται όμως στην περίπτωση που έχουμε παραπάνω από ένα ελάχιστα;

[ 12,13,5, -7, 12,13,18,0, -7, -5,-2,4,54, ]

Με αυτόν τον αλγόριθμο εκτυπώνουμε μόνο τη **πρώτη** θέση του ελαχίστου.

Αν επιθυμούμε να εκτυπώσουμε το **τελευταίο** ελάχιστο στοιχείο.

Αρκεί να προστεθεί ένα = στη συνθήκη του Αν.

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

### Πίνακες

1.23

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος  Εύρεση_ελαχίστου
  Δεδομένα  // N, A//
  ελάχιστος ← A[1]  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο
  θέση_ελαχίστου ← 1

  Για  i από 2 μέχρι N
    Αν  A[i] <= ελάχιστος τότε
      ελάχιστος ← A[i]
      θέση_ελαχίστου ← i
    Τέλος_αν
  Τέλος_επανάληψης
  Γράψε  "Το ελάχιστο είναι ο αριθμός", ελάχιστος
  Γράψε  "Η θέση του ελαχίστου είναι", θέση_ελαχίστου
  Τέλος  Εύρεση_ελαχίστου
  
```

Τι γίνεται όμως στην περίπτωση που έχουμε παραπάνω από ένα ελάχιστα;

[ 12,13,5, -7, 12,13,18,0, -7, -5,-2,4,54, ]

Με αυτόν τον αλγόριθμο εκτυπώνουμε μόνο τη **πρώτη** θέση του ελαχίστου. Αν επιθυμούμε να εκτυπώσουμε το **τελευταίο** ελάχιστο στοιχείο.

Αρκεί να προστεθεί ένα = στη συνθήκη του Αν.

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

### Πίνακες

1.23

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος  Εύρεση_ελαχίστου
Δεδομένα  // N, A//
ελάχιστος ← A[1]  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο
θέση_ελαχίστου ← 1

Για  i από 2 μέχρι N
    Αν A[i] <= ελάχιστος τότε
        ελάχιστος ← A[i]
        θέση_ελαχίστου ← i
    Τέλος_αν
Τέλος_επανάληψης
Γράψε  "Το ελάχιστο είναι ο αριθμός", ελάχιστος
Γράψε  "Η θέση του ελαχίστου είναι", θέση_ελαχίστου
Τέλος  Εύρεση_ελαχίστου
    
```

Τι γίνεται όμως στην περίπτωση που έχουμε παραπάνω από ένα ελάχιστα;

[ 12,13,5, -7, 12,13,18,0, -7, -5,-2,4,54, ]

Με αυτόν τον αλγόριθμο εκτυπώνουμε μόνο τη **πρώτη** θέση του ελαχίστου. Αν επιθυμούμε να εκτυπώσουμε το **τελευταίο** ελάχιστο στοιχείο. Αρκεί να προστεθεί ένα = στη συνθήκη του Αν.

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

### Πίνακες

1.23

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος  Εύρεση_ελαχίστου
Δεδομένα  // N, A//
ελάχιστος ← A[1]  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο
θέση_ελαχίστου ← 1

Για  i από 2 μέχρι N
    Αν  A[i] <= ελάχιστος τότε
        ελάχιστος ← A[i]
        θέση_ελαχίστου ← i
    Τέλος_αν
Τέλος_επανάληψης
Γράψε  "Το ελάχιστο είναι ο αριθμός", ελάχιστος
Γράψε  "Η θέση του ελαχίστου είναι", θέση_ελαχίστου
Τέλος  Εύρεση_ελαχίστου
    
```

Τι γίνεται όμως στην περίπτωση που έχουμε παραπάνω από ένα ελάχιστα;

[ 12,13,5, -7, 12,13,18,0, -7, -5,-2,4,54, ]

Με αυτόν τον αλγόριθμο εκτυπώνουμε μόνο τη **πρώτη** θέση του ελαχίστου. Αν επιθυμούμε να εκτυπώσουμε το **τελευταίο** ελάχιστο στοιχείο. Αρκεί να προστεθεί ένα = στη συνθήκη του Αν.

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

## Πίνακες

1.23

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος είναι ο ακόλουθος:

```

Αλγόριθμος  Εύρεση_ελαχίστου
Δεδομένα  // N, A//
ελάχιστος ← A[1]  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο
θέση_ελαχίστου ← 1

Για  i από 2 μέχρι N
    Αν  A[i] <= ελάχιστος τότε
        ελάχιστος ← A[i]
        θέση_ελαχίστου ← i
    Τέλος_αν
Τέλος_επανάληψης
Γράψε  "Το ελάχιστο είναι ο αριθμός", ελάχιστος
Γράψε  "Η θέση του ελαχίστου είναι", θέση_ελαχίστου
Τέλος  Εύρεση_ελαχίστου
    
```

Τι γίνεται όμως στην περίπτωση που έχουμε παραπάνω από ένα ελάχιστα;

[ 12,13,5, -7, 12,13,18,0, -7, -5,-2,4,54, ]

Με αυτόν τον αλγόριθμο εκτυπώνουμε μόνο τη **πρώτη** θέση του ελαχίστου. Αν επιθυμούμε να εκτυπώσουμε το **τελευταίο** ελάχιστο στοιχείο. Αρκεί να προστεθεί ένα = στη συνθήκη του Αν. **Αν όμως θέλουμε να εκτυπώσουμε όλες τις θέσεις των ελάχιστων:**

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος που θα εκτυπώνει όλα τα ελάχιστα στοιχεία είναι ο ακόλουθος:

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος που θα εκτυπώνει όλα τα ελάχιστα στοιχεία είναι ο ακόλουθος:

**Αλγόριθμος** Εύρεση\_όλα\_τα\_ελάχιστα

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος που θα εκτυπώνει όλα τα ελάχιστα στοιχεία είναι ο ακόλουθος:

**Αλγόριθμος** Εύρεση\_όλα\_τα\_ελάχιστα

**Δεδομένα** //  $N, A$

ελά;

**Τέλος** Εύρεση\_όλα\_τα\_ελάχιστα

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος που θα εκτυπώνει όλα τα ελάχιστα στοιχεία είναι ο ακόλουθος:

**Αλγόριθμος** Εύρεση\_όλα\_τα\_ελάχιστα

**Δεδομένα** //  $N, A$

ελάχιστος  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

**Για**  $i$  από 2 μ

**Τέλος** Εύρεση\_όλα\_τα\_ελάχιστα

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος που θα εκτυπώνει όλα τα ελάχιστα στοιχεία είναι ο ακόλουθος:

```
Αλγόριθμος  Εύρεση_όλα_τα_ελάχιστα
  Δεδομένα  // N, A//
  ελάχιστος ← A[1] ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

  Για  i από 2 μέχρι N
    Αν
      Τέλος  Εύρεση_όλα_τα_ελάχιστα
```

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος που θα εκτυπώνει όλα τα ελάχιστα στοιχεία είναι ο ακόλουθος:

**Αλγόριθμος** Εύρεση\_όλα\_τα\_ελάχιστα

**Δεδομένα** //  $N, A$

ελάχιστος  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

**Για**  $i$  από 2 μέχρι  $N$

**Αν**  $A[i] <$  ελάχιστος **τότε**  
ελάχιστος  $\leftarrow A[i]$

**Τέλος** Εύρεση\_όλα\_τα\_ελάχιστα

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος που θα εκτυπώνει όλα τα ελάχιστα στοιχεία είναι ο ακόλουθος:

**Αλγόριθμος** Εύρεση\_όλα\_τα\_ελάχιστα

**Δεδομένα** //  $N, A$

ελάχιστος  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

**Για**  $i$  από 2 μέχρι  $N$

**Αν**  $A[i] <$  ελάχιστος **τότε**

ελάχιστος  $\leftarrow A[i]$

**Τέλος\_αν**

**Τέλος** Εύρεση\_όλα\_τα\_ελάχιστα

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος που θα εκτυπώνει όλα τα ελάχιστα στοιχεία είναι ο ακόλουθος:

**Αλγόριθμος** Εύρεση\_όλα\_τα\_ελάχιστα

**Δεδομένα** //  $N, A$

**ελάχιστος**  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

**Για**  $i$  από 2 μέχρι  $N$

**Αν**  $A[i] < \text{ελάχιστος}$  τότε

**ελάχιστος**  $\leftarrow A[i]$

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Για**  $i$  από 1

**Τέλος** Εύρεση\_όλα\_τα\_ελάχιστα

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος που θα εκτυπώνει όλα τα ελάχιστα στοιχεία είναι ο ακόλουθος:

**Αλγόριθμος** Εύρεση\_όλα\_τα\_ελάχιστα

**Δεδομένα** //  $N, A$

**ελάχιστος**  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

**Για**  $i$  από 2 μέχρι  $N$

**Αν**  $A[i] < \text{ελάχιστος}$  τότε

**ελάχιστος**  $\leftarrow A[i]$

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Για**  $i$  από 1 μέχρι  $N$

**Αν**

**Τέλος** Εύρεση\_όλα\_τα\_ελάχιστα

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος που θα εκτυπώνει όλα τα ελάχιστα στοιχεία είναι ο ακόλουθος:

```
Αλγόριθμος  Εύρεση_όλα_τα_ελάχιστα
  Δεδομένα  // N, A//
  ελάχιστος ← A[1] ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

  Για  i από 2 μέχρι N
    Αν  A[i] < ελάχιστος τότε
      ελάχιστος ← A[i]
    Τέλος_αν
  Τέλος_επανάληψης

  Για  i από 1 μέχρι N
    Αν  A[i] = ελάχιστος τότε
      Γράψε
  Τέλος

Τέλος  Εύρεση_όλα_τα_ελάχιστα
```

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος που θα εκτυπώνει όλα τα ελάχιστα στοιχεία είναι ο ακόλουθος:

**Αλγόριθμος** Εύρεση\_όλα\_τα\_ελάχιστα

**Δεδομένα** //  $N, A$

**ελάχιστος**  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

**Για**  $i$  από 2 μέχρι  $N$

**Αν**  $A[i] < \text{ελάχιστος}$  τότε

**ελάχιστος**  $\leftarrow A[i]$

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Για**  $i$  από 1 μέχρι  $N$

**Αν**  $A[i] = \text{ελάχιστος}$  τότε

**Γράψε** "Η θέση του ελαχίστου είναι ",  $i$

**Τέλος\_αν**

**Τέλος** Εύρεση\_όλα\_τα\_ελάχιστα

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος που θα εκτυπώνει όλα τα ελάχιστα στοιχεία είναι ο ακόλουθος:

**Αλγόριθμος** Εύρεση\_όλα\_τα\_ελάχιστα

**Δεδομένα** //  $N, A$

**ελάχιστος**  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

**Για**  $i$  από 2 μέχρι  $N$

**Αν**  $A[i] < \text{ελάχιστος}$  τότε  
    **ελάχιστος**  $\leftarrow A[i]$

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Για**  $i$  από 1 μέχρι  $N$

**Αν**  $A[i] = \text{ελάχιστος}$  τότε

**Γράψε** "Η θέση του ελαχίστου είναι ",  $i$

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Τέλος** Εύρεση\_όλα\_τα\_ελάχιστα

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος που θα εκτυπώνει όλα τα ελάχιστα στοιχεία είναι ο ακόλουθος:

Αλγόριθμος **Εύρεση\_όλα\_τα\_ελάχιστα**

Δεδομένα //  $N, A$

ελάχιστος  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

Για  $i$  από 2 μέχρι  $N$

Αν  $A[i] <$  ελάχιστος τότε  
ελάχιστος  $\leftarrow A[i]$

Τέλος\_αν

Τέλος\_επανάληψης

Για  $i$  από 1 μέχρι  $N$

Αν  $A[i] =$  ελάχιστος τότε  
Γράψε "Η θέση του ελαχίστου είναι ",  $i$

Τέλος\_αν

Τέλος\_επανάληψης

Τέλος **Εύρεση\_όλα\_τα\_ελάχιστα**

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος που θα εκτυπώνει όλα τα ελάχιστα στοιχεία είναι ο ακόλουθος:

Αλγόριθμος **Εύρεση\_όλα\_τα\_ελάχιστα**

Δεδομένα //  $N, A$

ελάχιστος  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

Για  $i$  από 2 μέχρι  $N$

Αν  $A[i] <$  ελάχιστος τότε  
ελάχιστος  $\leftarrow A[i]$

Τέλος\_αν

Τέλος\_επανάληψης

Για  $i$  από 1 μέχρι  $N$

Αν  $A[i] =$  ελάχιστος τότε  
Γράψε "Η θέση του ελαχίστου είναι ",  $i$   
Τέλος\_αν

Τέλος\_επανάληψης

Τέλος **Εύρεση\_όλα\_τα\_ελάχιστα**



## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος που θα εκτυπώνει όλα τα ελάχιστα στοιχεία είναι ο ακόλουθος:

Αλγόριθμος Εύρεση\_όλα\_τα\_ελάχιστα

Δεδομένα //  $N, A$

ελάχιστος  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

Για  $i$  από 2 μέχρι  $N$

Αν  $A[i] <$  ελάχιστος τότε  
ελάχιστος  $\leftarrow A[i]$

Τέλος\_αν

Τέλος\_επανάληψης

Εδώ βρίσκουμε τη τιμή του ελάχιστου

Για  $i$  από 1 μέχρι  $N$

Αν  $A[i] =$  ελάχιστος τότε

Γράψε "Η θέση του ελαχίστου είναι ",  $i$

Τέλος\_αν

Τέλος\_επανάληψης

Τέλος Εύρεση\_όλα\_τα\_ελάχιστα

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος που θα εκτυπώνει όλα τα ελάχιστα στοιχεία είναι ο ακόλουθος:

Αλγόριθμος **Εύρεση\_όλα\_τα\_ελάχιστα**

Δεδομένα //  $N, A$

ελάχιστος  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

Για  $i$  από 2 μέχρι  $N$

Αν  $A[i] <$  ελάχιστος τότε  
ελάχιστος  $\leftarrow A[i]$

Τέλος\_αν

Τέλος\_επανάληψης

Εδώ βρίσκουμε τη  
τιμή του ελάχιστου

Για  $i$  από 1 μέχρι  $N$

Αν  $A[i] =$  ελάχιστος τότε

Γράψε "Η θέση του ελαχίστου είναι ",  $i$

Τέλος\_αν

Τέλος\_επανάληψης

Τέλος **Εύρεση\_όλα\_τα\_ελάχιστα**

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος που θα εκτυπώνει όλα τα ελάχιστα στοιχεία είναι ο ακόλουθος:

Αλγόριθμος **Εύρεση\_όλα\_τα\_ελάχιστα**

Δεδομένα //  $N, A$

ελάχιστος  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

Για  $i$  από 2 μέχρι  $N$

Αν  $A[i] <$  ελάχιστος τότε  
ελάχιστος  $\leftarrow A[i]$

Τέλος\_αν

Τέλος\_επανάληψης

Εδώ βρίσκουμε τη τιμή του ελάχιστου

Για  $i$  από 1 μέχρι  $N$

Αν  $A[i] =$  ελάχιστος τότε

Γράψε "Η θέση του ελαχίστου είναι ",  $i$

Τέλος\_αν

Τέλος\_επανάληψης

Τέλος **Εύρεση\_όλα\_τα\_ελάχιστα**



## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος που θα εκτυπώνει όλα τα ελάχιστα στοιχεία είναι ο ακόλουθος:

Αλγόριθμος ← Εύρεση\_όλα\_τα\_ελάχιστα

Δεδομένα //  $N, A$

ελάχιστος ←  $A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

Για  $i$  από 2 μέχρι  $N$

Αν  $A[i] <$  ελάχιστος τότε  
ελάχιστος ←  $A[i]$

Τέλος\_αν

Τέλος\_επανάληψης

Εδώ βρίσκουμε τη  
τιμή του ελάχιστου

Για  $i$  από 1 μέχρι  $N$

Αν  $A[i] =$  ελάχιστος τότε

Γράψε "Η θέση του ελαχίστου είναι ",  $i$

Τέλος\_αν

Τέλος\_επανάληψης

Τέλος Εύρεση\_όλα\_τα\_ελάχιστα

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος που θα εκτυπώνει όλα τα ελάχιστα στοιχεία είναι ο ακόλουθος:

Αλγόριθμος **Εύρεση\_όλα\_τα\_ελάχιστα**

Δεδομένα //  $N, A$

ελάχιστος  $\leftarrow A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

Για  $i$  από 2 μέχρι  $N$

Αν  $A[i] <$  ελάχιστος τότε  
ελάχιστος  $\leftarrow A[i]$

Τέλος\_αν

Τέλος\_επανάληψης

Εδώ βρίσκουμε τη  
τιμή του ελάχιστου

Για  $i$  από 1 μέχρι  $N$

Αν  $A[i] =$  ελάχιστος τότε

Γράψε "Η θέση του ελαχίστου είναι ",  $i$

Τέλος\_αν

Τέλος\_επανάληψης

Τέλος Εύρεση\_όλα\_τα\_ελάχιστα

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.23

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος με δεδομένο ένα μονοδιάστατο πίνακα  $N$  αριθμών, θα υπολογίζει και θα εκτυπώνει το ελάχιστο καθώς και τη θέση του στον πίνακα.

Ο αλγόριθμος που θα εκτυπώνει όλα τα ελάχιστα στοιχεία είναι ο ακόλουθος:

Αλγόριθμος ← Εύρεση\_όλα\_τα\_ελάχιστα

Δεδομένα //  $N, A$

ελάχιστος ←  $A[1]$  ! Έστω ότι το πρώτο στοιχείο είναι το ελάχιστο

Για  $i$  από 2 μέχρι  $N$

Αν  $A[i] <$  ελάχιστος τότε  
ελάχιστος ←  $A[i]$

Τέλος\_αν

Τέλος\_επανάληψης

Εδώ βρίσκουμε τη  
τιμή του ελάχιστου

Για  $i$  από 1 μέχρι  $N$

Αν  $A[i] =$  ελάχιστος τότε

Γράψε "Η θέση του ελαχίστου είναι ",  $i$

Τέλος\_αν

Τέλος\_επανάληψης

Εδώ εκτυπώνουμε  
τις θέσεις των  
ελάχιστων

Τέλος Εύρεση\_όλα\_τα\_ελάχιστα

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

## Πίνακες

1.24

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.24

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα, ο οποίος θα περιέχει μόνο τους θετικούς απ' αυτούς.

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

## Πίνακες

1.24

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.24

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα, ο οποίος θα περιέχει μόνο τους θετικούς απ' αυτούς.

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.24

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα, ο οποίος θα περιέχει μόνο τους θετικούς απ' αυτούς.

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.24

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα, ο οποίος θα περιέχει μόνο τους θετικούς απ' αυτούς.

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.24

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα, ο οποίος θα περιέχει μόνο τους θετικούς απ' αυτούς.

Πρόκειται για τη λειτουργία της αντιγραφής:

Όλοι ή

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.24

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα, ο οποίος θα περιέχει μόνο τους θετικούς απ' αυτούς.

Πρόκειται για τη λειτουργία της αντιγραφής:

Όλοι ή μερικοί κόμβοι μιας δομής αντιγράφονται σε μια άλλη.

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.24

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα, ο οποίος θα περιέχει μόνο τους θετικούς απ' αυτούς.

Πρόκειται για τη λειτουργία της αντιγραφής:

Όλοι ή  
μερικοί κόμβοι μιας δομής  
αντιγράφονται σε μια άλλη.

Όταν απαιτείται η δημιουργία νέου πίνακα κάνουμε τα εξής:

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.24

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα, ο οποίος θα περιέχει μόνο τους θετικούς απ' αυτούς.

Πρόκειται για τη λειτουργία της αντιγραφής:

Όλοι ή  
μερικοί κόμβοι μιας δομής  
αντιγράφονται σε μια άλλη.

Όταν απαιτείται η δημιουργία νέου πίνακα κάνουμε τα εξής:

Χρησιμοποιούμε μετρητή για τη καταμέτρηση των περιπτώσεων που μας ικανοποιούν,

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.24

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα, ο οποίος θα περιέχει μόνο τους θετικούς απ' αυτούς.

Πρόκειται για τη λειτουργία της αντιγραφής:

Όλοι ή  
μερικοί κόμβοι μιας δομής  
αντιγράφονται σε μια άλλη.

Όταν απαιτείται η δημιουργία νέου πίνακα κάνουμε τα εξής:

Χρησιμοποιούμε μετρητή για τη καταμέτρηση των περιπτώσεων που μας ικανοποιούν,  
και ο οποίος αποτελεί δείκτη του νέου

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.24

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα, ο οποίος θα περιέχει μόνο τους θετικούς απ' αυτούς.

Πρόκειται για τη λειτουργία της αντιγραφής:

Όλοι ή  
μερικοί κόμβοι μιας δομής  
αντιγράφονται σε μια άλλη.

Όταν απαιτείται η δημιουργία νέου πίνακα κάνουμε τα εξής:

Χρησιμοποιούμε μετρητή για τη καταμέτρηση των περιπτώσεων που μας ικανοποιούν,  
και ο οποίος αποτελεί δείκτη του νέου πίνακα που δημιουργείται.

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.24

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα, ο οποίος θα περιέχει μόνο τους θετικούς απ' αυτούς.

Ο αλγόριθμος είναι ο ακόλουθος:

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.24

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα, ο οποίος θα περιέχει μόνο τους θετικούς απ' αυτούς.

Ο αλγόριθμος είναι ο ακόλουθος:

Αλγόριθμος Θετικοί\_αρι

Τέλος

Θετικοί\_αρι

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.24

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα, ο οποίος θα περιέχει μόνο τους θετικούς απ' αυτούς.

Ο αλγόριθμος είναι ο ακόλουθος:

Αλγόριθμος Θετικοί\_αριθμοί

Δεδομένα //  $N, A$  //

Τέλος Θετικοί\_αριθμοί

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.24

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα, ο οποίος θα περιέχει μόνο τους θετικούς απ' αυτούς.

Ο αλγόριθμος είναι ο ακόλουθος:

Αλγόριθμος Θετικοί\_αριθμοί

Δεδομένα //  $N, A$  //

$M \leftarrow 0$

Τέλος Θετικοί\_αριθμοί

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.24

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα, ο οποίος θα περιέχει μόνο τους θετικούς απ' αυτούς.

Ο αλγόριθμος είναι ο ακόλουθος:

Αλγόριθμος Θετικοί\_αριθμοί

Δεδομένα //  $N, A$  //

$M \leftarrow 0$  ! δείκτης νέου πίνακα

Γι

Τέλος Θετικοί\_αριθμοί

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.24

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα, ο οποίος θα περιέχει μόνο τους θετικούς απ' αυτούς.

Ο αλγόριθμος είναι ο ακόλουθος:

Αλγόριθμος Θετικοί\_αριθμοί

Δεδομένα //  $N, A$  //

$M \leftarrow 0$  ! δείκτης νέου πίνακα

Για  $i$  από 1 μέχρι

Τέλος Θετικοί\_αριθμοί

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.24

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα, ο οποίος θα περιέχει μόνο τους θετικούς απ' αυτούς.

Ο αλγόριθμος είναι ο ακόλουθος:

Αλγόριθμος Θετικοί\_αριθμοί

Δεδομένα //  $N, A$  //

$M \leftarrow 0$  ! δείκτης νέου πίνακα

Για  $i$  από 1 μέχρι  $N$

Αν

Τέλος Θετικοί\_αριθμοί

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.24

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα, ο οποίος θα περιέχει μόνο τους θετικούς απ' αυτούς.

Ο αλγόριθμος είναι ο ακόλουθος:

Αλγόριθμος Θετικοί\_αριθμοί

Δεδομένα //  $N, A$  //

$M \leftarrow 0$  ! δείκτης νέου πίνακα

Για  $i$  από 1 μέχρι  $N$

Αν  $A[i] > 0$  τότε ! βρέθηκε άλλο ένα στοιχείο

Τέλος Θετικοί\_αριθμοί

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.24

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα, ο οποίος θα περιέχει μόνο τους θετικούς απ' αυτούς.

Ο αλγόριθμος είναι ο ακόλουθος:

Αλγόριθμος Θετικοί\_αριθμοί

Δεδομένα //  $N, A$  //

$M \leftarrow 0$  ! δείκτης νέου πίνακα

Για  $i$  από 1 μέχρι  $N$

Αν  $A[i] > 0$  τότε ! βρέθηκε άλλο ένα στοιχείο  
 $M \leftarrow M+1$

Τέλος Θετικοί\_αριθμοί



## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα, ο οποίος θα περιέχει μόνο τους θετικούς απ' αυτούς.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Θετικοί\_αριθμοί

**Δεδομένα** //  $N, A$  //

$M \leftarrow 0$  ! δείκτης νέου πίνακα

**Για**  $i$  από 1 μέχρι  $N$

**Αν**  $A[i] > 0$  τότε ! βρέθηκε άλλο ένα στοιχείο

$M \leftarrow M+1$  ! πήγαινε σε αυτή τη θέση

**Τέλος** Θετικοί\_αριθμοί



## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα, ο οποίος θα περιέχει μόνο τους θετικούς απ' αυτούς.

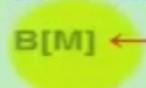
Ο αλγόριθμος είναι ο ακόλουθος:

```
Αλγόριθμος Θετικοί_αριθμοί
Δεδομένα // N, A//

M ← 0      ! δείκτης νέου πίνακα

Για i από 1 μέχρι N
  Αν A[i] > 0 τότε      ! βρέθηκε άλλο ένα στοιχείο
    M ← M+1           ! πήγαινε σε αυτή τη θέση
    B[M] ← A[i]

Τέλος Θετικοί_αριθμοί
```



## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα, ο οποίος θα περιέχει μόνο τους θετικούς απ' αυτούς.

Ο αλγόριθμος είναι ο ακόλουθος:

```
Αλγόριθμος Θετικοί_αριθμοί
Δεδομένα // N, A//

M ← 0      ! δείκτης νέου πίνακα

Για i από 1 μέχρι N
  Αν A[i] > 0 τότε      ! βρέθηκε άλλο ένα στοιχείο
    M ← M+1            ! πήγαινε σε αυτή τη θέση
    B[M] ← A[i]

Τέλος Θετικοί_αριθμοί
```

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.24

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα, ο οποίος θα περιέχει μόνο τους θετικούς απ' αυτούς.

Ο αλγόριθμος είναι ο ακόλουθος:

```
Αλγόριθμος Θετικοί_αριθμοί
Δεδομένα // N, A//
M ← 0      ! δείκτης νέου πίνακα
Για i από 1 μέχρι N
  Αν A[i] > 0 τότε      ! βρέθηκε άλλο ένα στοιχείο
    M ← M+1            ! πήγαινε σε αυτή τη θέση
    B[M] ← A[i]       ! και τοποθέτησε το εκεί
Τέλος Θετικοί_αριθμοί
```



## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.24

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα, ο οποίος θα περιέχει μόνο τους θετικούς απ' αυτούς.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Θετικοί\_αριθμοί

**Δεδομένα** //  $N, A$  //

$M \leftarrow 0$  ! δείκτης νέου πίνακα

**Για**  $i$  από 1 μέχρι  $N$

**Αν**  $A[i] > 0$  τότε ! βρέθηκε άλλο ένα στοιχείο

$M \leftarrow M+1$  ! πήγαινε σε αυτή τη θέση

$B[M] \leftarrow A[i]$  ! και τοποθέτησε το εκεί

**Τέλος** Θετικοί\_αριθμοί

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.24

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα, ο οποίος θα περιέχει μόνο τους θετικούς απ' αυτούς.

Ο αλγόριθμος είναι ο ακόλουθος:

Αλγόριθμος Θετικοί\_αριθμοί

Δεδομένα //  $N, A$  //

$M \leftarrow 0$  ! δείκτης νέου πίνακα

Για  $i$  από 1 μέχρι  $N$

Αν  $A[i] > 0$  τότε ! βρέθηκε άλλο ένα στοιχείο

$M \leftarrow M+1$  ! πήγαινε σε αυτή τη θέση

$B[M] \leftarrow A[i]$  ! και τοποθέτησε το εκεί

Τέλος\_αν

Τέλος\_επανάληψης

Τέλος Θετικοί\_αριθμοί

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα, ο οποίος θα περιέχει μόνο τους θετικούς απ' αυτούς.

Ο αλγόριθμος είναι ο ακόλουθος:

Αλγόριθμος Θετικοί\_αριθμοί

Δεδομένα //  $N, A$  //

$M \leftarrow 0$  ! δείκτης νέου πίνακα

Για  $i$  από 1 μέχρι  $N$

Αν  $A[i] > 0$  τότε ! βρέθηκε άλλο ένα στοιχείο

$M \leftarrow M+1$  ! πήγαινε σε αυτή τη θέση

$B[M] \leftarrow A[i]$  ! και τοποθέτησε το εκεί

Τέλος\_αν

Τέλος\_επανάληψης

Τέλος Θετικοί\_αριθμοί

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα, ο οποίος θα περιέχει μόνο τους θετικούς απ' αυτούς.

Ο αλγόριθμος είναι ο ακόλουθος:

```
Αλγόριθμος Θετικοί_αριθμοί
Δεδομένα // N, A//

M ← 0      ! δείκτης νέου πίνακα

Για i από 1 μέχρι N
  Αν A[i] > 0 τότε      ! βρέθηκε άλλο ένα στοιχείο
    M ← M+1           ! πήγαινε σε αυτή τη θέση
    B[M] ← A[i] ! και τοποθέτησε το εκεί
  Τέλος_αν
Τέλος_επανάληψης

Αποτελέσματα

Τέλος Θετικοί_αριθμοί
```

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.24

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα, ο οποίος θα περιέχει μόνο τους θετικούς απ' αυτούς.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Θετικοί\_αριθμοί

**Δεδομένα** //  $N, A$  //

$M \leftarrow 0$  ! δείκτης νέου πίνακα

**Για**  $i$  από 1 μέχρι  $N$

**Αν**  $A[i] > 0$  τότε ! βρέθηκε άλλο ένα στοιχείο

$M \leftarrow M+1$  ! πήγαινε σε αυτή τη θέση

$B[M] \leftarrow A[i]$  ! και τοποθέτησε το εκεί

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Αποτελέσματα** //  $M, B$  //

**Τέλος** Θετικοί\_αριθμοί

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος , με δεδομ

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

## Πίνακες

1.25

Να αναπτυχθεί αλγόριθμος ο οποίος , με δεδομένο έναν μονοδιάστατο πίνακα

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται  $n$

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η  $\xi$

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Παράδειγμα ο πίνακας:

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Παράδειγμα ο πίνακας:

[ 5 ]

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Παράδειγμα ο πίνακας:

 $[ 5, 0, 0, \dots ]$

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Παράδειγμα ο πίνακας:

 $[ 5, 0, 0, 9, 7, 0, ]$

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος , με δεδομένο έναν μονοδιάστατο πίνακα Ν αριθμών , θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία .Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Παράδειγμα ο πίνακας:

**[ 5 , 0 , 0 , 9 , 7 , 0 , -3 , 12 ]**

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Παράδειγμα ο πίνακας:

 $[ \quad , 0 , 0 , 9 , 7 , 0 , -3 , 12 ]$ 

θα γίνει

5

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Παράδειγμα ο πίνακας:

 $[ \quad , 0 , 0 , \quad , \quad , 0 , 12 ]$ 

θα γίνει

5 9 7 -3

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Παράδειγμα ο πίνακας:

[ , , , , , , , ]

θα γίνει

5 9 7 -3 12 0 0 0

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Παράδειγμα ο πίνακας:

[ , , , , , , , ]

θα γίνει

5, 9 7 -3 12 0 0 0

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Παράδειγμα ο πίνακας:

[ , , , , , , ]

θα γίνει

5, 9, 7, -3, 12 0 0 0

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Παράδειγμα ο πίνακας:

[ , , , , , , ]

θα γίνει

5, 9, 7, -3, 12, 0, 0, 0

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Παράδειγμα ο πίνακας:

[ , , , , , , , ]

θα γίνει

5, 9, 7, -3, 12, 0, 0, 0

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Παράδειγμα ο πίνακας:

 $[ 5, 0, 0, 9, 7, 0, -3, 12 ]$ 

θα γίνει

 $[ 5, 9, 7, -3, 12, 0, 0, 0 ]$

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Παράδειγμα ο πίνακας:

 $[ 5, 0, 0, 9, 7, 0, -3, 12 ]$ 

θα γίνει

 $[ 5, 9, 7, -3, 12, 0, 0, 0 ]$ 

Θα προσπελάσουμε τον πίνακα

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Παράδειγμα ο πίνακας:

$$[ 5, 0, 0, 9, 7, 0, -3, 12 ]$$


θα γίνει

$$[ 5, 9, 7, -3, 12, 0, 0, 0 ]$$

Θα προσπελάσουμε τον πίνακα και όταν εντοπίζουμε κάποιο μη μηδενικό στοιχείο

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

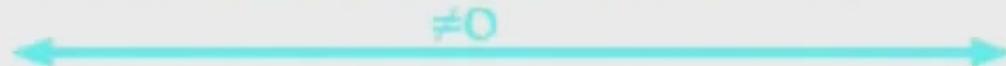
ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Παράδειγμα ο πίνακας:

$$[ 5, 0, 0, 9, 7, 0, -3, 12 ]$$


← ≠ 0 →

θα γίνει

$$[ 5, 9, 7, -3, 12, 0, 0, 0 ]$$

Θα προσπελάσουμε τον πίνακα

και όταν εντοπίζουμε κάποιο μη μηδενικό στοιχείο  
θα το αντιγράψουμε στο νέο πίνακα.

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Παράδειγμα ο πίνακας:

$$[ 5, 0, 0, 9, 7, 0, -3, 12 ]$$


$\neq 0$

θα γίνει

$$[ 5, 9, 7, -3, 12, 0, 0, 0 ]$$

Θα προσπελάσουμε τον πίνακα και όταν εντοπίζουμε κάποιο μη μηδενικό στοιχείο θα το αντιγράψουμε στο νέο πίνακα.

Κατόπιν θα τοπο

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Παράδειγμα ο πίνακας:

$$[ 5, 0, 0, 9, 7, 0, -3, 12 ]$$


θα γίνει

$$[ 5, 9, 7, -3, 12, 0, 0, 0 ]$$

Θα προσπελάσουμε τον πίνακα και όταν εντοπίζουμε κάποιο μη μηδενικό στοιχείο θα το αντιγράψουμε στο νέο πίνακα.

Κατόπιν θα τοποθετήσουμε την τιμή μηδέν σε όλες τις θέσεις που έχουν απομείνει.

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.25

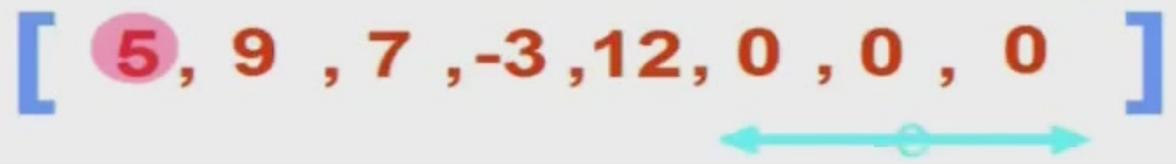
### Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Παράδειγμα ο πίνακας:



θα γίνει



Θα προσπελάσουμε τον πίνακα και όταν εντοπίζουμε κάποιο μη μηδενικό στοιχείο θα το αντιγράψουμε στο νέο πίνακα.

Κατόπιν θα τοποθετήσουμε την τιμή μηδέν σε όλες τις θέσεις που έχουν απομείνει.

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

c

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Ο αλγόριθμος είναι ο ακόλουθος:

Αλγόριθμος Διαχωρισμ

Τέ

Διαχωρισμ

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Διαχωρισμός\_μηδενικών

**Δεδομένα** //  $N, A$

**Τέλος** Διαχωρισμός\_μηδενικών

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Διαχωρισμός\_μηδενικών

**Δεδομένα** //  $N, A$

μη\_μηδέν ← 0

**Τέλος** Διαχωρισμός\_μηδενικών

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Διαχωρισμός\_μηδενικών

**Δεδομένα** //  $N, A$

μη\_μηδέν ← 0      ! δείκτης νέου πίνακα

**Για**

**Τέλος** Διαχωρισμός\_μηδενικών

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Διαχωρισμός\_μηδενικών

**Δεδομένα** //  $N, A$

μη\_μηδέν ← 0      ! δείκτης νέου πίνακα

**Για**  $i$  από 1 μέχρι

**Τέλος** Διαχωρισμός\_μηδενικών

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Διαχωρισμός\_μηδενικών

**Δεδομένα** //  $N, A$

μη\_μηδέν ← 0      ! δείκτης νέου πίνακα

**Για**  $i$  από 1 μέχρι  $N$

**Αν**

**Τέλος** Διαχωρισμός\_μηδενικών

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Διαχωρισμός\_μηδενικών

**Δεδομένα** //  $N, A$

μη\_μηδέν ← 0      ! δείκτης νέου πίνακα

**Για**  $i$  από 1 μέχρι  $N$

**Αν**  $A[i] <> 0$  τότε

μη\_μηδέν ← μη\_μηδέν + 1

**Τέλος** Διαχωρισμός\_μηδενικών

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Διαχωρισμός\_μηδενικών

**Δεδομένα** //  $N, A$

μη\_μηδέν  $\leftarrow 0$       ! δείκτης νέου πίνακα

**Για**  $i$  από 1 μέχρι  $N$

**Αν**  $A[i] <> 0$  τότε

μη\_μηδέν  $\leftarrow$  μη\_μηδέν + 1

**Τέλος** Διαχωρισμός\_μηδενικών

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Διαχωρισμός\_μηδενικών

**Δεδομένα** //  $N, A$

μη\_μηδέν  $\leftarrow 0$  ! δείκτης νέου πίνακα

**Για**  $i$  από 1 μέχρι  $N$

**Αν**  $A[i] \neq 0$  τότε

μη\_μηδέν  $\leftarrow$  μη\_μηδέν + 1

$B[\text{μη\_μηδέν}] \leftarrow A[i]$

! Μετρητής

**Τέλος** Διαχωρισμός\_μηδενικών

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Διαχωρισμός\_μηδενικών

**Δεδομένα** //  $N, A$

μη\_μηδέν ← 0      ! δείκτης νέου πίνακα

**Για**  $i$  από 1 μέχρι  $N$

**Αν**  $A[i] \neq 0$  τότε

μη\_μηδέν ← μη\_μηδέν + 1

$B[\text{μη\_μηδέν}] \leftarrow A[i]$

**Τέλος**

! Μετρητής

**Τέλος** Διαχωρισμός\_μηδενικών

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Διαχωρισμός\_μηδενικών

**Δεδομένα** //  $N, A$

μη\_μηδέν  $\leftarrow 0$  ! δείκτης νέου πίνακα

**Για**  $i$  από 1 μέχρι  $N$

**Αν**  $A[i] \neq 0$  τότε

μη\_μηδέν  $\leftarrow$  μη\_μηδέν + 1 ! Μετρητής

$B[\text{μη\_μηδέν}] \leftarrow A[i]$

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Τέλος** Διαχωρισμός\_μηδενικών

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Διαχωρισμός\_μηδενικών

**Δεδομένα** //  $N, A$

μη\_μηδέν  $\leftarrow 0$  ! δείκτης νέου πίνακα

**Για**  $i$  από 1 μέχρι  $N$

**Αν**  $A[i] \neq 0$  τότε

μη\_μηδέν  $\leftarrow$  μη\_μηδέν + 1 ! Μετρητής

$B[\text{μη\_μηδέν}] \leftarrow A[i]$

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Για**  $i$  από μη\_μηδέν + 1 μέχρι

**Τέλος** Διαχωρισμός\_μηδενικών

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Διαχωρισμός\_μηδενικών

**Δεδομένα** //  $N, A$

μη\_μηδέν ← 0 ! δείκτης νέου πίνακα

**Για**  $i$  από 1 μέχρι  $N$

**Αν**  $A[i] <> 0$  τότε

μη\_μηδέν ← μη\_μηδέν + 1 ! Μετρητής

$B[\text{μη\_μηδέν}] \leftarrow A[i]$

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Για**  $i$  από μη\_μηδέν + 1 μέχρι  $N$  ! τα υπόλοιπα

**Τέλος** Διαχωρισμός\_μηδενικών

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Διαχωρισμός\_μηδενικών

**Δεδομένα** //  $N, A$

μη\_μηδέν  $\leftarrow 0$  ! δείκτης νέου πίνακα

**Για**  $i$  από 1 μέχρι  $N$

**Αν**  $A[i] \neq 0$  τότε

μη\_μηδέν  $\leftarrow$  μη\_μηδέν + 1 ! Μετρητής

$B[\text{μη\_μηδέν}] \leftarrow A[i]$

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Για**  $i$  από μη\_μηδέν + 1 μέχρι  $N$  ! τα υπόλοιπα γίνονται μηδέν

$B[i] \leftarrow 0$

**Τέλος\_ε**

**Τέλος** Διαχωρισμός\_μηδενικών

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

## Πίνακες

1.25

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Διαχωρισμός\_μηδενικών

**Δεδομένα** //  $N, A$

μη\_μηδέν ← 0 ! δείκτης νέου πίνακα

**Για**  $i$  από 1 μέχρι  $N$

**Αν**  $A[i] \neq 0$  τότε

μη\_μηδέν ← μη\_μηδέν + 1 ! Μετρητής

$B[\text{μη\_μηδέν}] \leftarrow A[i]$

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Για**  $i$  από μη\_μηδέν + 1 μέχρι  $N$  ! τα υπόλοιπα γίνονται μηδέν

$B[i] \leftarrow 0$

**Τέλος\_επανάληψης**

**Αποτελέσματα**

**Τέλος** Διαχωρισμός\_μηδενικών

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Διαχωρισμός\_μηδενικών

**Δεδομένα** //  $N, A$

μη\_μηδέν ← 0 ! δείκτης νέου πίνακα

**Για**  $i$  από 1 μέχρι  $N$

**Αν**  $A[i] <> 0$  τότε

μη\_μηδέν ← μη\_μηδέν + 1 ! Μετρητής

$B[\text{μη\_μηδέν}] \leftarrow A[i]$

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Για**  $i$  από μη\_μηδέν + 1 μέχρι  $N$  ! τα υπόλοιπα γίνονται μηδέν

$B[i] \leftarrow 0$

**Τέλος\_επανάληψης**

**Αποτελέσματα** //  $N, B$

**Τέλος** Διαχωρισμός\_μηδενικών

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

## ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος Διαχωρισμός\_μηδενικών**

**Δεδομένα** //  $N, A$

μη\_μηδέν ← 0 ! δείκτης νέου πίνακα

**Για**  $i$  από 1 μέχρι  $N$

**Αν**  $A[i] <> 0$  τότε

μη\_μηδέν ← μη\_μηδέν + 1 ! Μετρητής

$B[\text{μη\_μηδέν}] \leftarrow A[i]$

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Για**  $i$  από μη\_μηδέν + 1 μέχρι  $N$  ! τα υπόλοιπα γίνονται μηδέν

$B[i] \leftarrow 0$

**Τέλος\_επανάληψης**

**Αποτελέσματα** //  $N, B$

**Τέλος Διαχωρισμός\_μηδενικών**

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Διαχωρισμός\_μηδενικών

**Δεδομένα** //  $N, A$

μη\_μηδέν ← 0 ! δείκτης νέου πίνακα

**Για**  $i$  από 1 μέχρι  $N$

**Αν**  $A[i] <> 0$  τότε

μη\_μηδέν ← μη\_μηδέν + 1 ! Μετρητής

$B[\text{μη\_μηδέν}] \leftarrow A[i]$

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Για**  $i$  από μη\_μηδέν + 1 μέχρι  $N$  ! τα υπόλοιπα γίνονται μηδέν

$B[i] \leftarrow 0$

**Τέλος\_επανάληψης**

**Αποτελέσματα** //  $N, B$

**Τέλος** Διαχωρισμός\_μηδενικών

## ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΕΠΑΝΑΛΗΨΗ

1.25

## Πίνακες

Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένο έναν μονοδιάστατο πίνακα  $N$  αριθμών, θα δημιουργεί νέο πίνακα όπου θα έχουν διαχωριστεί τα μη μηδενικά στοιχεία με τα μηδενικά στοιχεία. Προσοχή! Δε θα αλλοιώνεται η διάταξη των αριθμών.

Ο αλγόριθμος είναι ο ακόλουθος:

**Αλγόριθμος** Διαχωρισμός\_μηδενικών

**Δεδομένα** //  $N, A$

μη\_μηδέν  $\leftarrow 0$  ! δείκτης νέου πίνακα

**Για**  $i$  από 1 μέχρι  $N$

**Αν**  $A[i] <> 0$  τότε

μη\_μηδέν  $\leftarrow$  μη\_μηδέν + 1 ! Μετρητής

$B[\text{μη\_μηδέν}] \leftarrow A[i]$

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Για**  $i$  από μη\_μηδέν + 1 μέχρι  $N$  ! τα υπόλοιπα γίνονται μηδέν

$B[i] \leftarrow 0$

**Τέλος\_επανάληψης**

**Αποτελέσματα** //  $N, B$

**Τέλος** Διαχωρισμός\_μηδενικών

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

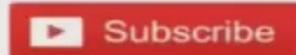
 Σπύρος Γ. Ζυγούρης  
Καθηγητής Πληροφορικής

 **spzygouris@gmail.com**

You **Tube**



Spyros Georgios Zygoris

 Subscribe