



ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
ΑΝΟΙΚΤΑ ΑΚΑΔΗΜΑΪΚΑ ΜΑΘΗΜΑΤΑ



ΥΠΟΛΟΓΙΣΤΕΣ II

Μονοδιάστατοι πίνακες

Διδάσκοντες: **Αν. Καθ. Δ. Παπαγεωργίου,**
Αν. Καθ. Ε. Λοιδωρικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



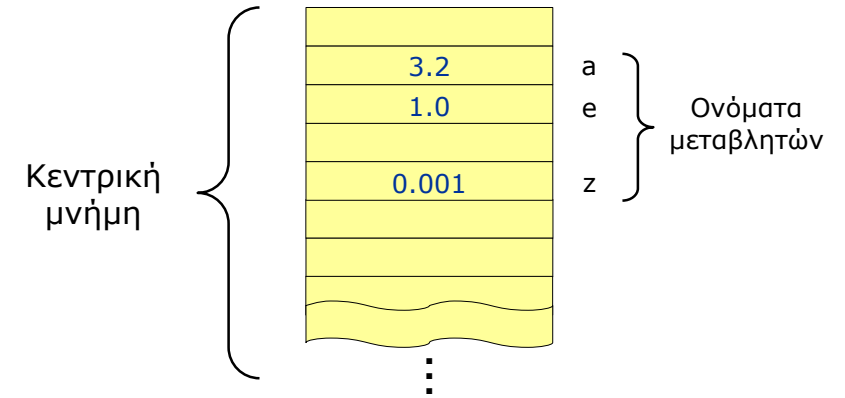
ΥΠΟΛΟΓΙΣΤΕΣ ΙΙ

Μονοδιάστατοι πίνακες

1

Τι είναι οι πίνακες;

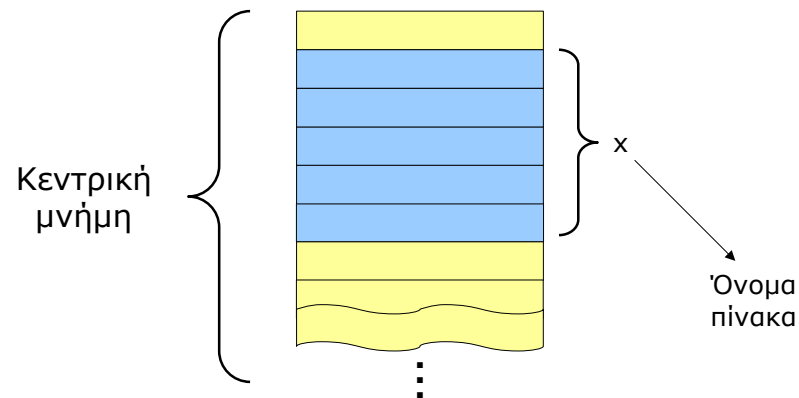
Απλές μεταβλητές:



2

Τι είναι οι πίνακες;

Πίνακες:



3

Τι είναι οι πίνακες;

Μια ομάδα διαδοχικών θέσεων αποθήκευσης στην κεντρική μνήμη, στην οποία αναφερόμαστε με **ένα κοινό όνομα**.

Κάθε στοιχείο του πίνακα ξεχωρίζει από τα υπόλοιπα με τη χρήση ενός **δείκτη**. Έτσι για ένα πίνακα που ονομάζεται x:

Το πρώτο στοιχείο είναι το $x[0]$

Το δεύτερο στοιχείο είναι το $x[1]$

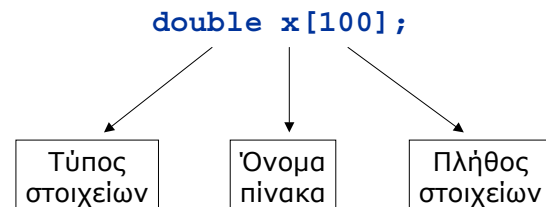
Σημείωση:

Όλα τα στοιχεία του πίνακα είναι του ίδιου τύπου.

4

Δήλωση

Πριν χρησιμοποιήσουμε ένα πίνακα πρέπει να τον δηλώσουμε. Για να δηλώσουμε ένα πίνακα που ονομάζεται *x* και αποτελείται από 100 στοιχεία διπλής ακρίβειας γράφουμε:



5

Ορισμένες παρατηρήσεις

- Το 100 είναι το πλήθος των στοιχείων του πίνακα και όχι ο δείκτης του τελευταίου στοιχείου.
- Τα στοιχεία του πίνακα είναι τα: `x[0], x[1], x[2], ... x[99]`.
- Το κάθε στοιχείο του πίνακα μπορεί να χρησιμοποιηθεί όπως και μια μεταβλητή του αντίστοιχου τύπου. Πχ.
`x[0] = 4;`
`a = 3*x[0]+2*x[1];`
`cin >> x[4];`
- Από τα 100 στοιχεία μπορούμε να χρησιμοποιήσουμε στο πρόγραμμα λιγότερα.

6

Δήλωση

Το πλήθος των στοιχείων μπορεί να δηλωθεί με τη χρήση συμβολικών σταθερών:

```
#define NMAX 100
double x[NMAX];
```

Η γραμμή `#define` απευθύνεται στον προεπεξεργαστή και ορίζει μια συμβολική σταθερά - **όχι μεταβλητή** -

Προσοχή: όχι ερωτηματικό στο τέλος του `#define`

7

Ανάθεση τιμών στα στοιχεία πίνακα

Ανάθεση τιμών κατά τη δήλωση:

```
int a[10]={4,7,1,-3,0};
```

Η ανωτέρω δήλωση αναθέτει αρχικές τιμές στα 5 πρώτα από τα 10 στοιχεία του πίνακα *a*

8

Ανάθεση τιμών στα στοιχεία πίνακα

Απευθείας ανάθεση:

```
x[0] = 1.2;
x[1] = 2.5;
x[2] = z+2*x[0];

for ( k=0; k<n; ++k )
    x[k] = k*k/2;
```

9

Ανάθεση τιμών στα στοιχεία πίνακα

Εισαγωγή από το πληκτρολόγιο:

Εισάγουμε ένα-προς-ένα τα στοιχεία:

```
#define NMAX 100
double x[NMAX];

cout << "Πόσα στοιχεία ? ";
cin >> n;
if ( n>NMAX || n<1 ) {
    cout << "Λανθασμένος αριθμός \n";
    return 1;
}
for ( k=0; k<n; ++k ) {
    cout << "Εισάγετε το στοιχείο " << k << endl;
    cin >> x[k];
}
```

10

Εμφάνιση των στοιχείων πίνακα στην οθόνη

Εμφανίζουμε ένα-προς-ένα τα στοιχεία:

```
#define NMAX 100
double x[NMAX];

for ( k=0; k<n; ++k )
    cout << x[k] << endl;
```

11

Παράδειγμα #1

Βρείτε το μέσο όρο μιας σειράς αριθμών που εισάγονται από το πληκτρολόγιο.

12

Παράδειγμα #1

```
#include <iostream>
using namespace std;

#define NMAX 100
int main ( )
{
    double x[NMAX];
    int n, k;
    double s, avg;

    cout << "Πόσα στοιχεία ? ";
    cin >> n;
    if ( n > NMAX || n < 1 ) {
        cout << "Λάθος \n";
        return 1;
    }
    for ( k=0; k<n; ++k) {
        cout << "Εισάγετε το στοιχείο " << k << endl;
        cin >> x[k];
    }
}
```

Συνεχίζεται...

13

Παράδειγμα #1

...Συνέχεια

```
s = 0;
for ( k=0; k<n; ++k )
    s += x[k];
avg = s/n;

cout << "Ο μέσος όρος είναι: " << avg << endl;
}
```

14

Παράδειγμα #2

Συμπληρώστε το προηγούμενο παράδειγμα έτσι ώστε να εμφανίζει στην οθόνη πόσο απέχει κάθε αριθμός που εισάγαμε από το μέσο όρο.

15

Παράδειγμα #2

```
for ( k=0; k<n; ++k )
    cout << k << "    " << x[k] << "    "
        << abs(avg-x[k]) << endl;
```

Τυπικό αποτέλεσμα στην οθόνη για 4 στοιχεία:

0	1.7	0.8805
1	0.85	1.7305
2	3.6	1.0195
3	4.172	1.5915

Παρατηρείστε ότι οι τρεις στήλες των αποτελεσμάτων δεν είναι στοιχισμένες.

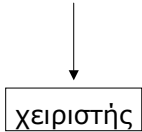
16

Μορφοποίηση της εξόδου

Για αν μορφοποιήσουμε κατά βούληση την εμφάνιση των αποτελεσμάτων μας στην οθόνη χρησιμοποιούμε τους **χειριστές** στην εντολή `cout`.

Πχ. για να εμφανίσουμε έναν ακέραιο ώστε να καταλαμβάνει 5 θέσεις στην οθόνη:

```
cout << setw(5) << k << endl;
```



17

Οι βασικότεροι χειριστές

- **setw(n)**
Η επόμενη έξοδος θα καταλαμβάνει n θέσεις στην οθόνη.
- **setprecision(n)**
Όλες οι επόμενες εξοδοι θα γίνουν με n δεκαδικά ψηφία.
- **fixed**
Όλες οι επόμενες εξοδοι θα γίνουν με την κλασσική αναπαράσταση.
- **scientific**
Όλες οι επόμενες εξοδοι θα γίνουν με την επιστημονική αναπαράσταση.

Σημείωση: Για να χρησιμοποιήσουμε τους χειριστές μορφοποίησης πρέπει να δηλώσουμε:

```
#include <iomanip>
```

18

Παράδειγμα #2 (με μορφοποίηση)

```
cout << fixed << setprecision(7);
for ( k=0; k<n; ++k )
    cout << setw(3) << k << "    "
        << setw(10) << x[k] << "    "
        << setw(10) << abs(avg-x[k]) << endl;
```

Τυπικό αποτέλεσμα στην οθόνη για 4 στοιχεία:

0	1.7000000	0.8805000
1	0.8500000	1.7305000
2	3.6000000	1.0195000
3	4.1720000	1.5915000

19

Παράδειγμα #3

Βρείτε το μικρότερο στοιχείο ενός πίνακα με n στοιχεία.

20

Παράδειγμα #3

```
min = x[0];
for ( k=1; k<n; ++k)
    if ( x[k] < min )
        min = x[k];
```

21

Παράδειγμα #4

Βρείτε το μικρότερο στοιχείο ενός πίνακα με η στοιχεία και τη θέση στην οποία βρίσκεται αυτό.

22

Παράδειγμα #4

```
min = x[0];
kmin = 0;
for ( k=1; k<n; ++k)
    if ( x[k] < min ) {
        min = x[k];
        kmin = k;
    }
```

23

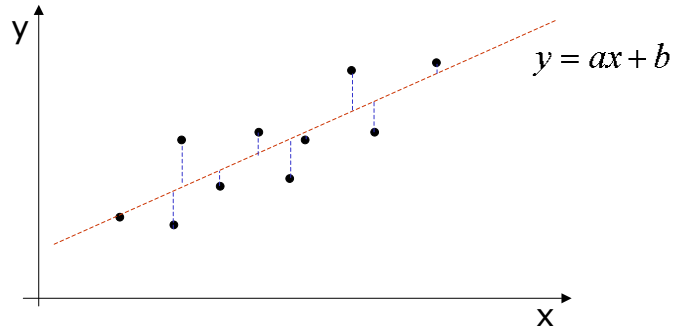
Παράδειγμα #5

Προσαρμογή ελαχίστων τετραγώνων

Δεδομένων N σημείων (x_i, y_i) στο επίπεδο προσδιορίστε τη βέλτιστη ευθεία που ελαχιστοποιεί τις αποστάσεις από τα σημεία.

24

Παράδειγμα #5



25

Παράδειγμα #5

Η βέλτιστη ευθεία έχει εξίσωση: $y = ax + b$

Οι συντελεστές a , b δίνονται από:

$$a = \frac{Ns_{xy} - s_x s_y}{Ns_{xx} - s_x s_x} \quad b = \frac{s_{xx} s_y - s_x s_{xy}}{Ns_{xx} - s_x s_x}$$

$$s_x = \sum_{i=1}^N X_i \quad s_y = \sum_{i=1}^N Y_i \quad s_{xx} = \sum_{i=1}^N X_i^2 \quad s_{xy} = \sum_{i=1}^N X_i Y_i$$

26

Παράδειγμα #5

```
#include <iostream>
using namespace std;
#define NMAX 100

int main ( )
{
    double x[NMAX], y[NMAX];
    int n, k;
    double sx, sy, sxx, sxy;
    double a, b;

    cout << "Πόσα σημεία έχετε ";
    cin >> n;
    if ( n < 1 || n > NMAX ) {
        cout << "Λάθος \n";
        return 1;
    }
    for ( k=0; k<n; ++k ) {
        cout << "Δώστε το σημείο " << k << endl;
        cin >> x[k] >> y[k];
    }
}
```

Συνεχίζεται...

27

Παράδειγμα #5

```

sx = sy = sxx = sxy = 0;
for ( k=0; k<n; ++k ) {
    sx += x[k];
    sy += y[k];
    sxx += x[k]*x[k];
    sxy += x[k]*y[k];
}

a = (n*sxy - sx*sy) / (n*sxx - sx*sx);
b = (sxx*sy - sx*sxy) / (n*sxx - sx*sx);

cout << "Οι συντελεστές είναι: \n";
cout << a << endl;
cout << b << endl;
}

```

...Συνέχεια

28

Παράδειγμα #6

Σύστημα φορτισμένων σωματιδίων

Θεωρήστε ένα σύστημα που αποτελείται από N φορτισμένα σωματίδια, το καθένα με φορτίο q_i , και μάζα m_i . Η θέση του κάθε σωματιδίου προσδιορίζεται από τις συντεταγμένες x_i, y_i, z_i .

Βρείτε:

- το ολικό φορτίο του συστήματος
- το κέντρο μάζας του συστήματος
- την ολική ηλεκτροστατική ενέργεια

29

Παράδειγμα #6

Το ολικό φορτίο είναι: $q_{tot} = \sum q_i$

Το κέντρο μάζας είναι:

$$x_m = \frac{\sum x_i m_i}{\sum m_i} \quad y_m = \frac{\sum y_i m_i}{\sum m_i} \quad z_m = \frac{\sum z_i m_i}{\sum m_i}$$

Η ηλεκτροστατική ενέργεια είναι:

$$E = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \frac{q_i q_j}{|r_i - r_j|}$$

30

Παράδειγμα #6 (Δηλώσεις)

```
#include <iostream>
#include <cmath>
using namespace std;

#define NMAX 100           // Μέγιστο πλήθος σωματιδίων

int main ( )
{
    int n;                 // Πλήθος σωματιδίων
    double x[NMAX], y[NMAX], z[NMAX]; // Συντεταγμένες
    double m[NMAX], q[NMAX]; // Μάζες και φορτία
    int k, i, j;
    double sx, sy, sz, sm;
    double xm, ym, zm;    // Κέντρο μάζας
    double qtot;          // Ολικό φορτίο
    double e;             // Ηλεκτρ. ενέργεια.
    double d;
```

Συνεχίζεται...

31

Παράδειγμα #6 (Εισαγωγή)

```
...Συνέχεια

cout << "Πόσα σωματίδια? \n";
cin >> n;
if ( n < 1 || n > NMAX ) {
    cout << "Λανθασμένος αριθμός\n";
    return 1;
}
for ( k=0; k<n; ++k ) {
    cout << "Δώστε συντεταγμένες, μάζα και φορτίο "
         << "για το σωματίδιο " << k << endl;
    cin >> x[k] >> y[k] >> z[k] >> m[k] >> q[k];
}

Συνεχίζεται...
```

32

Παράδειγμα #6 (Ολικό φορτίο)

...Συνέχεια

```
qtot = 0;
for ( k=0; k<n; ++k )
    qtot += q[k];
cout << "Το ολικό φορτίο είναι " << qtot << endl;
```

Συνεχίζεται...

33

Παράδειγμα #6 (Κέντρο μάζας)

...Συνέχεια

```
sx = sy = sz = sm = 0;
for ( k=0; k<n; ++k ) {
    sx += m[k]*x[k];
    sy += m[k]*y[k];
    sz += m[k]*z[k];
    sm += m[k];
}
xm = sx/sm;
ym = sy/sm;
zm = sz/sm;
cout << "Το κέντρο μάζας είναι "
    << xm << " " << ym << " " << zm << endl;
```

Συνεχίζεται...

34

Παράδειγμα #6 (Ηλ. ενέργεια)

...Συνέχεια

```
e = 0;
for ( i=0; i<n-1; ++i )
    for ( j=i+1; j<n; ++j ) {
        d = sqrt( pow(x[i]-x[j],2)+pow(y[i]-y[j],2)+
                pow(z[i]-z[j],2) );
        e += q[i]*q[j]/d;
    }
cout << "Η ηλεκτροστατική ενέργεια είναι "
    << e << endl;
```

35

ΥΠΟΛΟΓΙΣΤΕΣ ΙΙ

Πίνακες χαρακτήρων

36

Μεταβλητές τύπου *char*

- Που χρησιμεύουν:
Για την αποθήκευση μεμονωμένων χαρακτήρων.
- Δήλωση:
`char c;`
- Εισαγωγή από το πληκτρολόγιο:
`cin >> c;`
- Εμφάνιση στην οθόνη:
`cout << c;`
- Ανάθεση τιμής:

```
c = 'A';           // Ο χαρακτήρας A
c = '4';           // Ο χαρακτήρας 4
c = '\n';          // Ο χαρακτήρας αλλαγής γραμμής
c = '\0';          // Ο χαρακτήρας με αριθμό 0
```

37

Πίνακες χαρακτήρων (συμβολοσειρές)

- Που χρησιμεύουν:
Για την αποθήκευση σειρών χαρακτήρων (λέξεων, φράσεων, κειμένου).
- Δήλωση:
`#define MAX 100`
`char s[MAX];`
- Εισαγωγή από το πληκτρολόγιο (μόνο μια λέξη):
`cin >> s;`
- Εμφάνιση στην οθόνη:
`cout << s;`

38

Χαρακτήρας τερματισμού

Από το μέγιστο δηλωμένο μέγεθος ενός πίνακα χαρακτήρων μπορεί να χρησιμοποιήσουμε λιγότερες θέσεις. Μετά το τέλος των χρησιμοποιημένων θέσεων τοποθετείται ο χαρακτήρας τερματισμού. Πχ για ένα πίνακα 10 χαρακτήρων:

Θέσεις: 0 1 2 3 4 5 6 7 8 9
 Περιεχόμενο:

H	E	L	L	O					
---	---	---	---	---	--	--	--	--	--



Χαρακτήρας τερματισμού
 Χαρακτήρας που αντιστοιχεί στον ακέραιο 0
 Συμβολίζεται: `\0`

39

Ποιος τοποθετεί το χαρακτήρα τερματισμού ;

Εξαρτάται πως δίνω τιμή στη συμβολοσειρά:

- `cin >> s;`
Ο χαρακτήρας τερματισμού τοποθετείται **αυτόματα** από την `cin`.
- Μπορώ να δώσω τιμές με απευθείας ανάθεση (δεν συνιστάται):

```
s[0] = 'H';
s[1] = 'E';
s[2] = 'L';
s[3] = 'L';
s[4] = 'O';
s[5] = '\0';
```

40

Παράδειγμα #7

Κατασκευάστε πρόγραμμα που θα δέχεται ως είσοδο μια συμβολοσειρά και θα βρίσκει το μήκος της.

Παρατήρηση:

Θα πρέπει να βρούμε σε ποια θέση του πίνακα βρίσκεται ο χαρακτήρας τερματισμού.

41

Παράδειγμα #7

```
#include <iostream>
using namespace std;

#define MAX 100

int main ( )
{
    char s[MAX];
    int k;

    cout << "Δώστε μια λέξη ";
    cin >> s;

    for ( k=0; s[k] != '\0'; ++k )
        ;

    cout << "Το μήκος είναι " << k << endl;
}
```

42

Συναρτήσεις για συμβολοσειρές

- `strlen (s)`
Επιστρέφει το πλήθος των χαρακτήρων της συμβολοσειράς s (Δεν συμπεριλαμβάνεται ο χαρακτήρας τερματισμού).
- `strcpy (s1, s2)`
Αντιγράφει τη συμβολοσειρά s2 στη s1.
- `strcmp (s1, s2)`
Συγκρίνει τις συμβολοσειρές s1 και s2 και επιστρέφει:
-1 αν η s1 είναι λεξικογραφικά μικρότερη από την s2.
0 αν η s1 είναι λεξικογραφικά ίση με την s2.
1 αν η s1 είναι λεξικογραφικά μεγαλύτερη από την s2.

Για να χρησιμοποιήσουμε τις ανωτέρω συναρτήσεις για συμβολοσειρές πρέπει να βάλουμε:

```
#include <cstring>
```

43

Συναρτήσεις για συμβολοσειρές

- `gets (s)`
Εισαγωγή από το πληκτρολόγιο της συμβολοσειράς s (μια ολόκληρη φράση μέχρι να πατήσουμε Enter). Τοποθετεί αυτόματα και το χαρακτήρα τερματισμού.

Για να χρησιμοποιήσουμε την ανωτέρω συνάρτηση πρέπει να βάλουμε:

```
#include <cstdio>
```

44

Παράδειγμα #8

Κατασκευάστε πρόγραμμα που θα δέχεται ως είσοδο μια συμβολοσειρά και θα μετατρέπει τυχόν κεφαλαίους χαρακτήρες σε μικρούς.

45

Παράδειγμα #8

Πως μετατρέπουμε ένα χαρακτήρα από κεφαλαίο σε μικρό;

Βασίζομαστε στο ότι:

- Υπάρχει αντιστοιχία χαρακτήρων αριθμών.
- Στον πίνακα αντιστοίχισης υπάρχουν τρεις ομάδες χαρακτήρων: κεφαλαία, μικρά, ψηφία.
- Οι χαρακτήρες κάθε ομάδας βρίσκονται σε διαδοχικές θέσεις του πίνακα.

Ακέραιος	Χαρακτήρας
48	0
49	1
50	2
51	3
52	4
...	...
65	A
66	B
67	C
68	D
69	E
...	...
97	a
98	b
99	c
100	d
101	e
...	...

46

Παράδειγμα #8

Για να μετατρέψουμε ένα χαρακτήρα (`char c`) από κεφαλαίο σε μικρό:

- Βρίσκουμε πόσο απέχει από την αρχή των κεφαλαίων (δηλαδή από το A):
`d = c - 'A';`
- Προσθέτουμε την απόσταση αυτή στην αρχή των μικρών (δηλαδή στο a):
`new = 'a' + d;`
- Ή συνολικά:
`new = 'a' - 'A' + c;`

Στις αριθμητικές πράξεις χρησιμοποιούμε χαρακτήρες. Εκεί εννοείται ο αντίστοιχος ακέραιος.

47

Παράδειγμα #8

Πως αποφασίζουμε αν ένας χαρακτήρας (`char c`) είναι κεφαλαίος ;

Πρέπει να βρίσκεται μεταξύ των χαρακτήρων A και Z

```
if ( c >= 'A' && c <= 'Z' )
```

48

Παράδειγμα #8

```
#include <iostream>
#include <cstdio>
#include <cstring>
using namespace std;

#define MAX 100

main ( )
{
    char s[MAX] ;
    char snw[MAX];
    int len, k;

    cout << "Εισάγετε μια συμβολοσειρά ";
    gets(s);
```

Συνεχίζεται...

49

Παράδειγμα #8

```
...Συνέχεια

    len = strlen(s);
    for (k=0; k<len; ++k)
        if (s[k]>='A' && s[k]<='Z')
            snw[k] = s[k] - 'A' + 'a';
        else
            snw[k] = s[k];

    snw[len] = '\0';
    cout << snw << endl;
}
```

50

Παράδειγμα #9

Κατασκευάστε πρόγραμμα που θα δέχεται ως είσοδο μια συμβολοσειρά και θα μετράει πόσοι:

- 1) Κεφαλαίοι χαρακτήρες
- 2) Μικροί χαρακτήρες
- 3) Αριθμητικά ψηφία
- 4) Σύμβολα

υπάρχουν στη συμβολοσειρά.

51

Παράδειγμα #9

```
#include <iostream>
#include <cstdio>
#include <cstring>
using namespace std;

#define MAX 100

int main ( )
{
    char str[MAX];
    int len;
    int n1;
    int n2;
    int n3;
    int n4;
    int k;

    // Μήκος αλφαριθμητικού
    // Πλήθος κεφαλαίων
    // Πλήθος μικρών
    // Πλήθος ψηφίων
    // Πλήθος σύμβολων

    cout << "Εισάγετε μια συμβολοσειρά ";
    gets(str);
```

Συνεχίζεται...

52

Παράδειγμα #9

...Συνέχεια

```
n1 = n2 = n3 = n4 = 0;
len = strlen(str);
for (k=0; k<len; ++k)
    if (str[k]>='A' && str[k]<='Z')
        ++n1;
    else if (str[k]>='a' && str[k]<='z')
        ++n2;
    else if (str[k]>='0' && str[k]<='9')
        ++n3;
    else
        ++n4;

cout << "Δώσατε: " << str << endl;
cout << n1 << " Κεφαλαία \n";
cout << n2 << " Μικρά \n";
cout << n3 << " Ψηφία \n";
cout << n4 << " Σύμβολα \n";
}
```

Τέλος Ενότητας



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Ιωαννίνων**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «**Εκπαίδευση και Δια Βίου Μάθηση**» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
Περίοδος προγραμματισμού 2007-2013

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
Μίσια για το μέλλον
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Σημειώματα

Σημείωμα Ιστορικού Εκδόσεων Έργου

Το παρόν έργο αποτελεί την έκδοση 1.0.

Έχουν προηγηθεί οι κάτωθι εκδόσεις:

- Έκδοση 1.0 διαθέσιμη εδώ.

<http://ecourse.uoi.gr/course/view.php?id=1227>.

Σημείωμα Αναφοράς

Copyright Πανεπιστήμιο Ιωαννίνων, Διδάσκοντες: Αν.
Καθ. Δ. Παπαγεωργίου, Αν. Καθ. Ε. Λοιδωρίκης.
«ΥΠΟΛΟΓΙΣΤΕΣ ΙΙ. Μονοδιάστατοι πίνακες». Έκδοση:
1.0. Ιωάννινα 2014. Διαθέσιμο από τη δικτυακή
διεύθυνση:
<http://ecourse.uoi.gr/course/view.php?id=1227>.

Σημείωμα Αδειοδότησης

- Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά Δημιουργού - Παρόμοια Διανομή, Διεθνής Έκδοση 4.0 [1] ή μεταγενέστερη.



- [1] <https://creativecommons.org/licenses/by-sa/4.0/>.