Θεματική ενότητα 7: Πίνακες – αλφαριθμητικά

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Πίνακες

- **Ο πίνακας είναι μία συλλογή μεταβλητών ίδιου τύπου, οι οποίες είναι αποθηκευμένες σε διαδοχικές θέσεις μνήμης. Χρησιμοποιείται για την αποθήκευση και διαχείριση δεδομένων κοινού τύπου και αποτελεί, μαζί με τους δείκτες, από τα πλέον ισχυρά εργαλεία της γλώσσας C.**
- **Δήλωση: float temp[31]**

Προσδιορίζει τον αριθμό των στοιχείων του πίνακα

Προσδιορίζει τον τύπο των στοιχείων του πίνακα

Προσδιορίζει το όνομα του πίνακα

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

- **Η δήλωση γνωστοποιεί στο μεταγλωττιστή ότι δηλώνεται ο πίνακας temp, ο οποίος περιέχει δεδομένα τύπου float και καταλαμβάνει 31 θέσεις μνήμης για την αποθήκευση των στοιχείων του.**
- **Αναφορά σε στοιχείο πίνακα: γίνεται με συνδυασμό του ονόματος και ενός δείκτη που εκφράζει τη σειρά του στοιχείου μέσα στον πίνακα:**

  temp[0]: ***πρώτο στοιχείο του πίνακα***

  temp[1]: ***δεύτερο στοιχείο του πίνακα***

  temp[30]: ***τελευταίο (τριακοστό πρώτο) στοιχείο του πίνακα***

- **Απόδοση αρχικής τιμής: κατά τη δήλωση του πίνακα, γίνεται με χρήση του τελεστή ανάθεσης ως εξής:**

float temp[5] = {1,2,-4.2,6,8}: αρχικοποιούνται και τα 5 στοιχεία του πίνακα temp.

float temp[5] = {1,2,-4.2}: αρχικοποιούνται τα 3 πρώτα στοιχεία του πίνακα temp, δηλαδή τα temp[0], temp[1], temp[2].

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

- **Παρατήρηση:** Όταν αποδίδονται αρχικές τιμές μπορεί να παραληφθεί το μέγεθος του πίνακα. Ο υπολογιστής θα υπολογίσει αυτόματα πόσα είναι τα στοιχεία του πίνακα από τον αριθμό των αρχικών τιμών που δίδονται. Η δήλωση: `char d_ar[] = {'a', 'b', 'c', 'd'};` έχει ως αποτέλεσμα τη δημιουργία ενός πίνακα χαρακτήρων (*char*) τεσσάρων στοιχείων με αρχικές τιμές:

`d_ar[0] = 'a'` `d_ar[1] = 'b'` `d_ar[2] = 'c'` `d_ar[3] = 'd'`

Στο πρότυπο της γλώσσας C99 έχουν εισαχθεί οι ***καθοριστές (designators)***, οι οποίοι επιτρέπουν τη στοχευμένη αρχικοποίηση στοιχείων σε έναν πίνακα. Είναι ακέραιοι αριθμοί και λειτουργούν ως αριθμοδείκτες, καθώς γράφονται μέσα σε αγκύλες και καθορίζουν τη θέση στον πίνακα στην οποία θα τοποθετηθεί η αρχική τιμή. Η πρόταση `float array[5]={[2]=1.7, [4]=-19.3};` δημιουργεί τον πίνακα array και αρχικοποιεί το τρίτο και το πέμπτο στοιχείο του πίνακα με τις τιμές 1.7 και -19.3, αντίστοιχα.

Πανεπιστήμιο Δυτικής Αττικής | Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Ένα επιπρόσθετο πλεονέκτημα των καθοριστών είναι ότι μπορούν να τοποθετηθούν με οποιαδήποτε σειρά κι όχι αποκλειστικά με αύξουσα, όπως φαίνεαι παρακάτω:

```
float array[9]={[2]=1.7, [7]=-19.3, [5]=13};
```

Εάν το μέγεθος ενός πίνακα έχει ορισθεί με τη δήλωση, π.χ. array[9], ο καθοριστές μπορούν να λάβουν τιμή από 0 έως και 8. Εάν όμως δεν ορισθεί το μέγεθος του πίνακα, καθορίζεται έμμεσα από τη μέγιστη τιμή καθοριστή, ο οποίος δεν πρέπει να είναι αρνητικός ακέραιος. Στην ακόλουθη πρόταση:

```
float array[]={[2]=1.7, [15]=-19.3, [5]=13};
```

το μέγεθος του πίνακα καθορίζεται στο 16 (15+1).

Πανεπιστήμιο Δυτικής Αττικής | Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

- Το γεγονός ότι η δείκτες των στοιχείων ενός πίνακα ξεκινούν από το 0 κι όχι από το 1 μπορεί αρχικά να προκαλέσει σύγχυση αλλά αντανακλά τη φιλοσοφία της C, η οποία επιδιώκει να παραμείνει ο προγραμματισμός κοντά στην αρχιτεκτονική του υπολογιστή. Το 0 αποτελεί το σημείο εκκίνησης για τους υπολογιστές. Εάν η αρίθμηση των στοιχείων πίνακα ξεκινούσε από το 1, όπως π.χ. FORTRAN, ο μεταγλωττιστής θα έπρεπε να αφαιρέσει 1 από κάθε αναφορά σε δείκτη στοιχείου για να ληφθεί η πραγματική διεύθυνση ενός στοιχείου. Επομένως, η επιλογή της C παράγει πιο αποτελεσματικό κώδικα.

- **Προσοχή**: Υπάρχει διαφορά ανάμεσα στη *δήλωση πίνακα* και στην *αναφορά στοιχείου σε πίνακα*. Σε μία δήλωση, ο δείκτης καθορίζει το μέγεθος του πίνακα. Σε μία αναφορά στοιχείου πίνακα, ο δείκτης προσδιορίζει το στοιχείο του πίνακα στο οποίο αναφερόμαστε. Π.χ. στη δήλωση `int temp[31];` το 31 δηλώνει τον αριθμό των στοιχείων του πίνακα. Αντίθετα στη `temp[13]=21;` το 13 δηλώνει το συγκεκριμένο στοιχείο ($14^{ο}$) του πίνακα, στο οποίο αναφερόμαστε και αποδίδουμε την τιμή 21.

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Ανάγνωση και εκτύπωση πίνακα

Η ανάγνωση και εκτύπωση ενός πίνακα γίνονται κατά στοιχείο, με τους κανόνες που ισχύουν για κάθε τύπο δεδομένου:

```
for ( i=0;i<arrayaySize;i++ )
{
   scanf( "%f",&array[i] );
   printf( "array[%d]=%f",i,array[i] );
}
```

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Παράδειγμα μονοδιάστατου πίνακα

Να γραφεί πρόγραμμα, το οποίο θα δέχεται από το πληκτρολόγιο ένα θετικό ακέραιο αριθμό **n** ψηφίων $x = x_{n-1}x_{n-2}...x_1x_0$, θα αποθηεύει τα ψηφία του $x_k, k = 0,1,...,n-1$ σε πίνακα και θα εμφανίζει στην οθόνη τον αριθμό με αντεστραμμένα τα ψηφία του, δηλαδή τον αριθμό $y = y_{n-1}y_{n-2}...y_1y_0 = x_0x_1...x_{n-2}x_{n-1}$.

```
#include <stdio.h>
#define N 7
int main()   {
    int x,y,array[N],z[N],i;
    printf( "Give a %d digit positive integer: ",N );
    scanf( "%d",&x );
```

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

```
    z[0]=1;
    for (i=1;i<N;i++)   z[i]=z[i-1]*10;
    y=x;
    for (i=(N-1);i>=1;i—)    {
        array[i]=y/z[i];
        y=y%z[i];
    }
    array[0]=y;
    y=array[N-1] ;
    for (i=1;i<N;i++) y=y+z[i]*array[N-(i+1)];
    printf( "\nx=%d, y=%d\n",x,y );
    return 0 ;
}
```

```
Give a 7 digit positive integer:  1234567

x=1234567, y=7654321
```

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Πολυδιάστατοι πίνακες

- ***Πίνακες, τα στοιχεία των οποίων είναι επίσης πίνακες*. Η πρόταση int array[4][12]; δηλώνει τη μεταβλητή array ως πίνακα 4 στοιχείων, όπου το κάθε στοιχείο είναι πίνακας 12 στοιχείων ακεραίων.**
- **Η C δε θέτει περιορισμό στον αριθμό των διαστάσεων των πινάκων.**
- **Αν και ο πολυδιάστατος πίνακας αποθηκεύεται στη μνήμη ως μία ακολουθία στοιχείων μίας διάστασης, μπορούμε να το θεωρούμε ως πίνακα πινάκων. Για παράδειγμα, έστω το επόμενο «μαγικό τετράγωνο», του οποίου οι γραμμές οριζόντια, κάθετα και διαγώνια δίνουν το ίδιο άθροισμα:**

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

17	24	1	8	15
23	5	7	14	16
4	6	13	20	22
10	12	19	21	3
11	18	25	2	9

Για να αποθηκευθεί το τετράγωνο αυτό σε πίνακα θα μπορούσαμε να κάνουμε την ακόλουθη δήλωση:

```
int magic[5][5]={ {17, 24, 1, 8, 15},
                  {23, 5, 7, 14, 16},
                  {4, 6, 13, 20, 22},
                  {10, 12, 19, 21, 3},
                  {11, 18, 25, 2, 9}
                };
```

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

- Από τον προηγούμενο κώδικα γίνεται αντιληπτό ότι στην απόδοση των αρχικών τιμών οι τιμές των στοιχείων κάθε γραμμής είναι κλεισμένες σε άγκιστρα.

- Για την αναφορά σε στοιχείο ενός πολυδιάστατου πίνακα θα πρέπει να καθορισθούν τόσοι δείκτες όσοι είναι αναγκαίοι. Έτσι, η έκφραση magic[1] αναφέρεται στη δεύτερη γραμμή του πίνακα, ενώ η έκφραση magic[1][3] αναφέρεται στο τέταρτο στοιχείο της δεύτερης γραμμής του πίνακα.

- Οι πολυδιάστατοι πίνακες αποθηκεύονται κατά γραμμές, που σημαίνει ότι ο τελευταίος δείκτης θέσης μεταβάλλεται ταχύτερα κατά την προσπέλαση των στοιχείων. Για παράδειγμα, ο πίνακας που δηλώνεται ως:

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

```
int ar[2][3]={ {0, 1, 2},
               {3, 4, 5}
             };
```

αποθηκεύεται όπως φαίνεται ακολούθως:

Στοιχείο	Μνήμη Διεύθυνση	Περιεχόμενα
ar[0] [0]	1000	0
ar[0] [1]	1004	1
ar [0] [2]	1008	2
ar[1] [0]	100C	3
ar[1] [1]	1010	4
ar[1] [2]	1014	5
	1018	

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Αρχικοποίηση πολυδιάστατου πίνακα

Για την αρχικοποίηση ενός πολυδιάστατου πίνακα *κλείνουμε κάθε γραμμή αρχικών τιμών σε άγκιστρα*. Αν δεν υπάρχουν οι αναγκαίες αρχικές τιμές, τα επιπλέον στοιχεία λαμβάνουν αρχική τιμή 0. Έτσι, στη δήλωση και αρχικοποίηση του ακόλουθου πίνακα

```
int ar[5][3]={ {1, 2, 3},
               {4},
               {5, 6, 7}
             };
```

η μεταβλητή ar δηλώνεται ως πίνακας 5 γραμμών και 3 στηλών. Ωστόσο έχουν αποδοθεί τιμές μόνο για τις 3 πρώτες γραμμές του πίνακα και μάλιστα για τη δεύτερη γραμμή έχει ορισθεί η τιμή μόνο του πρώτου στοιχείου. Η παραπάνω δήλωση έχει ως αποτέλεσμα τη δημιουργία του ακόλουθου πίνακα:

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

1 2 3

4 0 0

5 6 7

0 0 0

0 0 0

Αν δε συμπεριληφθούν τα ενδιάμεσα άγκιστρα:

```
int ar[5][3]={ 1, 2, 3,
               4,
               5, 6, 7 };
```

το αποτέλεσμα είναι ο ακόλουθος πίνακας, που είναι σαφώς διαφορετικός, οπότε δημιουργείται πρόβλημα:

1 2 3

4 5 6

7 0 0

0 0 0

0 0 0

Πανεπιστήμιο Δυτικής Αττικής | Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Προσοχή: Όπως και με τους πίνακες μίας διάστασης, έτσι και στους πολυδιάστατους πίνακες εαν αμεληθεί να δοθεί το μέγεθος του πίνακα, ο μεταγλωττιστής θα το καθορίσει αυτόματα με βάση τον αριθμό αρχικών τιμών που παρουσιάζονται. Ωστόσο στους πολυδιάστατους πίνακες μπορεί να παραληφθεί ο αριθμός των στοιχείων μόνο της πρώτης διάστασης, καθώς ο μεταγλωττιστής μπορεί να τον υπολογίσει από τον αριθμό των αρχικών τιμών που διατίθενται. Η παρακάτω δήλωση αξιοποιεί τη δυνατότητα αυτή του μεταγλωττιστή:

```
int ar[][3][2]={ {{1, 1}, {0, 0}, {1, 1}},
                 {{0, 0}, {1, 2}, {0, 1}}
               };
```

Με την παραπάνω δήλωση ο ar δηλώνεται αυτόματα ως πίνακας 2x3x2.

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

- **Η παρακάτω δήλωση:**

```
int ar[][]={ 1, 2, 3, 4, 5, 6};
```

είναι ανεπίτρεπτη επειδή ο μεταγλωττιστής δεν μπορεί να γνωρίζει τι είδους θα ήταν αυτός ο πίνακας. Θα μπορούσε να το θεωρήσει είτε πίνακα 2x3 είτε 3x2.

- **Η παρακάτω δήλωση:**

```
printf( "%d",array[1,2] );
```

είναι λανθασμένη στη γλώσσα C αλλά δεν εντοπίζεται από το μεταγλωττιστή και οδηγεί σε ανεπιθύμητα αποτελέσματα. Η σωστή είναι

```
printf( "%d",array[1][2] );
```

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Αποθήκευση των πινάκων στη μνήμη

Η πρόταση δήλωσης int ar[5]; έχει ως αποτέλεσμα τη δέσμευση χώρου στη μνήμη για την αποθήκευση 5 μεταβλητών ακέραιου τύπου. Έστω ότι μέσα στον κώδικα του προγράμματος υπάρχουν οι παρακάτω προτάσεις ανάθεσης, που αποδίδουν τιμές σε στοιχεία του πίνακα:

```
ar[0] = 18;
ar[2] = 21;
ar[4] = ar[0] + 12;
```

Στο σχήμα που ακολουθεί φαίνεται η μορφή που έχει η μνήμη που έχει δεσμευτεί για την αποθήκευση του πίνακα ar , θεωρώντας 32 bits για κάθε ακέραιο και πως ο υπολογιστής αρχίζει στη διεύθυνση 1000.

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Στοιχείο	Διεύθυνση	Περιεχόμενα (4 bytes)
	0FFC	
ar[0]	1000	**18**
ar[1]	1004	**απροσδιόριστη**
ar[2]	1008	**21**
ar[3]	100C	**απροσδιόριστη**
ar[4]	1010	**30**
	1014	

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

- Γίνεται αντιληπτό ότι τα ar[1] και ar[3] έχουν απροσδιόριστες τιμές. Τα περιεχόμενα αυτών των θέσεων μνήμης είναι ο,τιδήποτε έχει μείνει στις θέσεις αυτές από προηγούμενη αποθήκευση. Οι απροσδιόριστες τιμές αναφέρονται ως «σκουπίδια» (junk) και πολλές φορές αποτελούν αιτία πρόκλησης λαθών.

- Για την αποφυγή προβλημάτων αυτής της μορφής πρέπει να αποδίδονται αρχικές τιμές στους πίνακες ή να υπάρχει συνεχής αντίληψη του τι περιλαμβάνει ένας πίνακας. Θα πρέπει να σημειωθεί ότι η ANSI C διασφαλίζει ότι οι καθολικές μεταβλητές (αρχικοποιούνται από το σύστημα με την τιμή 0. Αυτό συμβαίνει βέβαια και για όλα τα στοιχεία πίνακα που έχει δηλωθεί ως καθολική μεταβλητή.

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Παράδειγμα 1: **Θεωρούμε πίνακα που αναπαριστά τις μέσες θερμοκρασίες των μηνών των τελευταίων τριών ετών. Να δοθούν: *(α)* βρόχος για τον υπολογισμό των μέσων ετήσιων θερμοκρασιών των τριών ετών, *(β)* βρόχος για τον υπολογισμό των μέσων μηνιαίων θερμοκρασιών των τριών ετών.**

(α)

```
#define YEARS 3
#define MONTHS 12
int year, month;
float subtotal, temp[YEARS][MONTHS], avg_temp[YEARS];
for (year=0;year<YEARS;year++) {   //ο temp θεωρείται αρχικοποιημένος
  subtotal=0.0;
  for (month=0;month<MONTHS;month++)
          subtotal+=temp[year][month];
  avg_temp[year]=subtotal/MONTHS;
}
```

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

(β)

```
#define YEARS 3
#define MONTHS 12
int year, month;
float subtotal, temp[YEARS][MONTHS], avg_temp[MONTHS];
//ο temp θεωρείται αρχικοποιημένος
for (month=0;month<MONTHS;month++)
{
  subtotal=0.0;
  for (year=0;year<YEARS;year++)
        subtotal+=temp[year][month];
  avg_temp[month]=subtotal/YEARS;
}
```

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Γενικευμένο παράδειγμα πράξεων με πίνακες

arrays_operations.c

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Παράδειγμα 2: πολυδιάστατος πίνακας

Σε μία εταιρεία εργάζονται 5 πωλητές. Οι συνολικές μηνιαίες αποδοχές κάθε πωλητή προκύπτουν από το άθροισμα του βασικού μισθού (700 €) και του 9% επί του μηνιαίου τζίρου που πέτυχε ο πωλητής τον εκάστοτε μήνα. Π.χ. εάν ο μηνιαίος τζίρος που πέτυχε ένας πωλητής είναι 12000€, τότε οι αποδοχές του αυτό το μήνα θα είναι $700 + 0.09 \cdot 12000 = 1780$ €.

Με βάση τα παραπάνω, να γραφεί πρόγραμμα, το οποίο θα επιτελεί τα ακόλουθα:

- Θα δημιουργεί πίνακα αριθμών κινητής υποδιαστολής, διαστάσεων **5x3**.
- Με χρήση κατάλληλης επαναληπτικής πρότασης θα δίνεται από το πληκτρολόγιο ο μηνιαίος τζίρος κάθε πωλητή και ακολούθως θα υπολογίζονται οι μηνιαίες αποδοχές, οι οποίες θα αποθηκεύονται στον ανωτέρω πίνακα και θα εμφανίζονται στην οθόνη. Ταυτόχρονα θα υπολογίζεται ο συνολικός τζίρος και θα εμφανίζεται στην οθόνη.

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Παράδειγμα 2: πολυδιάστατος πίνακας

- Ο τζίρος κάθε πωλητή και το ποσοστό που συνεισφέρει κάθε πωλητής στον συνολικό τζίρο θα αποθηκεύονται στον ανωτέρω πίνακα. Τα ποσοστά θα εμφανίζονται στην οθόνη.

- Θα υπολογίζεται και θα εμφανίζεται στην οθόνη ο αριθμός των πωλητών, για τους οποίους οι μηνιαίες αποδοχές εντάσσονται σε κάποια από τις ακόλουθες κατηγορίες.

I. 700-1500€

II. 1501-2000€

III. περισσότερα από 2000€

```
#include <stdio.h>
#define N 5
#define BS 700
#define COEF 0.09
#define FC 1500
#define SC 1500
```

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Παράδειγμα 2: πολυδιάστατος πίνακας

```
int main()
{
  float salesman[N][3],gross=0.0;
  int i,cat[3]={0,0,0};
  for (i=0;i<N;i++)
  {
    printf( "\nSalesman no %d, give the gross sales:  ",i+1 );
    scanf( "%.2f",&salesman[i][1] );
    salesman[i][0]=BS+COEF*salesman[i][1];
    gross=gross+salesman[i][1];
  }
  printf( "\nTotal gross sales: %.2f",gross );
```

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Παράδειγμα 2: πολυδιάστατος πίνακας

```
for (i=0;i<N;i++)
{
  printf( "\nSalesman no %d, salary: %.2f",i+1,salesman[i][0] );
  salesman[i][2]=100*salesman[i][1]/gross;
  printf(" \nSalesman no %d contribution to the total gross sales: );
  printf(" %.2f\n",i+1,salesman[i][2] );
  if (salesman[i][0]>2000) cat[2]++;
  else if (salesman[i][0]>1500) cat[1]++;
  else cat[0]++;
}
```

Πανεπιστήμιο Δυτικής Αττικής, Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Παράδειγμα 2: πολυδιάστατος πίνακας

```
for (i=0;i<3;i++)
{
  if (cat[i]==0)
          printf( "\nNo salesman falls into category %d\n",i+1 );
  else if (cat[i]==1)
          printf( "\n1 salesman falls into category %d\n",i+1 );
  else printf( "\n%d salesmen fall into category %d\n",cat[i],i+1 );
}
  return 0;
}
```

Πανεπιστήμιο Δυτικής Αττικής, Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Παράδειγμα 2: πολυδιάστατος πίνακας

```
Salesman no 1, give the gross sales: 12000

Salesman no 2, give the gross sales: 15000

Salesman no 3, give the gross sales: 17000

Salesman no 4, give the gross sales: 30000

Salesman no 5, give the gross sales: 25000

Total gross sales: 99000.00
Salesman no 1, salary: 1780.00
Salesman no 1 contribution to the total gross sales: 12.12

Salesman no 2, salary: 2050.00
Salesman no 2 contribution to the total gross sales: 15.15

Salesman no 3, salary: 2230.00
Salesman no 3 contribution to the total gross sales: 17.17

Salesman no 4, salary: 3400.00
Salesman no 4 contribution to the total gross sales: 30.30

Salesman no 5, salary: 2950.00
Salesman no 4 contribution to the total gross sales: 25.25

No salesman falls into category 1

1 salesman falls into category 2

4 salesmen fall into category 3
```

Πανεπιστήμιο Δυτικής Αττικής, Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Πίνακες μεταβλητού μήκους

Έως τώρα η δήλωση ενός πίνακα προϋπέθετε ότι οι διαστάσεις τους θα ήταν εκ των προτέρων γνωστές, δηλαδή κατά τον χρόνο μεταγλώττισης του προγράμματος. Αυτός είναι ένας σημαντικός περιορισμός, καθώς σε πλήθος εφαρμογών οι διαστάσεις ενός πίνακα *προκύπτουν* κατά την εκτέλεση του προγράμματος (π.χ. ως αποτέλεσμα της εκτέλεσης μίας πρότασης **if–else**) .

Στο πρότυπο C99 της γλώσσας C οι πίνακες ενισχύθηκαν με τη δυνατότητα να δηλώνονται οι διαστάσεις τους κατά τον χρόνο εκτέλεσης του προγράμματος. Οι πίνακες αυτοί ονομάζονται πίνακες μεταβλητού μήκους (variable length arrays, VLA). Σημειώνεται ότι οι διαστάσεις τους καθορίζονται άπαξ, όπως στους κλασσικούς πίνακες, και δεν μπορούν να μεταβληθούν. Μία τέτοια δυνατότητα δίνει η δυναμική διαχείριση μνήμης, που θα παρουσιαστεί αργότερα.

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Πίνακες ως παράμετροι συναρτήσεων

Εάν κατά την κλήση μίας συνάρτησης η πραγματική παράμετρος είναι όνομα πίνακα (πχ. **arr**), δεν αποστέλλεται στη συνάρτηση ολόκληρος ο πίνακας αλλά μόνο η διεύθυνση μνήμης του πρώτου byte του πρώτου στοιχείου του πίνακα. Η παράμετρος στη δήλωση της συνάρτησης είναι ένα όνομα τοπικού πίνακα (π.χ. **localArr**), ο οποίος κρατά ένα αντίγραφο της ίδιας διεύθυνσης. Με τον τρόπο αυτό η μεταβλητή **localArr** στην πραγματικότητα διαχειρίζεται τις θέσεις μνήμης που καταλαμβάνει ο πίνακας της καλούσας συνάρτησης, κατά συνέπεια μπορεί να μεταβάλλει με έμμεσο τρόπο τις τιμές του πίνακα και αυτές οι μεταβολές να διατηρηθούν και μετά την ολοκλήρωση της κλήσης της συνάρτησης.

Επιπρόσθετα, στη δήλωση δεν αναφέρεται το ακριβές μέγεθός του αλλά μόνο το γεγονός ότι είναι πίνακας καθώς και ο τύπος στοιχείων του. Επομένως ουσιαστικά γνωστοποιείται στον μεταγλωττιστή η διεύθυνση του πρώτου byte και η απόσταση (αριθμός bytes) μεταξύ των στοιχείων. Ο μηχανισμός κλήσης συνάρτησης με όρισμα πίνακα στηρίζεται στην έννοια των δεικτών και θα μελετηθεί εκτενώς σε επόμενη ενότητα.

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Παράδειγμα 4

Να γραφεί πρόγραμμα, το οποίο θα λαμβάνει 4 ακέραιους από το πληκτρολόγιο και θα τους αποδίδει σε πίνακα ακεραίων (**array[4]**). Στη συνέχεια θα καλείται η συνάρτηση **void pwr(int localArray[], int arraySize)**, η οποία θα μεταβάλλει τις τιμές των στοιχείων του πίνακα, υψώνοντας κάθε τιμή στοιχείου του πίνακα array στο τετράγωνο. Η **main()** θα τελειώνει με την εμφάνιση των νέων τιμών του **array** στην οθόνη.

```
#include <stdio.h>
#include <stdlib.h>
void pwr (int localArray[], int arraySize);
int main()  {
 int array[4],i;
 for (i=0;i<4;i++)
 {
          printf( "\narray[%d]= ",i );
          scanf( "%d",&array[i] );
 }
  printf( "\n\nInitial array:\n" );
for (i=0;i<4;i++) printf( "\t%d\n",array[i] );
pwr(array,4);
printf( "\n\nFinal array:\n" );
for (i=0;i<4;i++) printf( "\t%d\n",array[i] );
return 0;
}
```

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Παράδειγμα 4

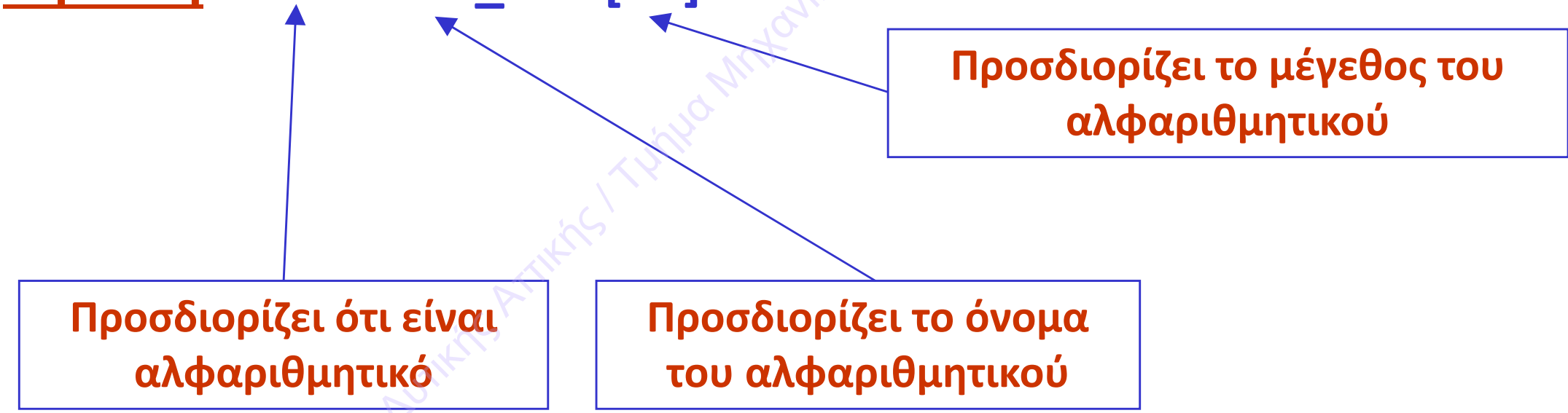
```
void pwr (int localArray[], int arraySize)
{
    int i;
    for  (i=0;i<4;i++)  localArray[i]=localArray[i]*localArray[i];

}
```

```
array[0]= 12

array[0]= 11

array[0]= 6

array[0]= 8

Initial array:
        12
        11
        6
        8

Final array:
        144
        121
        36
        64
```

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Το αλφαριθμητικό (string)

- Πίνακας χαρακτήρων που τερματίζει με το μηδενικό (*null*) χαρακτήρα. Ο μηδενικός χαρακτήρας έχει ASCII κωδικό 0 και αναπαρίσταται από την ακολουθία διαφυγής \0.
- Δήλωση: char book_title[30]



Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Αλφαριθμητική σταθερά

Τα αλφαριθμητικά μπορούν να εμφανίζονται μέσα στον κώδικα όπως οι αριθμητικές σταθερές, αποτελώντας τις αλφαριθμητικές σταθερές. Η αλφαριθμητική σταθερά απαντήθηκε σε προηγούμενο στάδιο, όπου και σημειώθηκε ότι οι χαρακτήρες της περικλείονται σε διπλά εισαγωγικά. Για την αποθήκευσή της χρησιμοποιείται ένας πίνακας χαρακτήρων, με τον μεταγλωττιστή να θέτει αυτόματα στο τέλος του αλφαριθμητικού ένα μηδενικό χαρακτήρα για να προσδιορίσει το τέλος του. Έτσι, η αλφαριθμητική σταθερά "Hello" απαιτεί για αποθήκευση 6 bytes, όπως φαίνεται παρακάτω:

'H' 'e' 'l' 'l' 'o' '\0'

***Προσοχή*: Σημειώστε τη διαφορά ανάμεσα στη σταθερά χαρακτήρα 'A' και την αλφαριθμητική σταθερά "A". Η πρώτη απαιτεί 1 byte για αποθήκευση, ενώ η δεύτερη ένα για το χαρακτήρα A κι ένα για το null.**

Πανεπιστήμιο Δυτικής Αττικής, Σχολή Μηχανικών Πληροφορικής και Υπολογιστών

Απόδοση αρχικής τιμής σε αλφαριθμητικό

Η ανάθεση τιμής με τη δήλωση ακολουθεί το γενικό κανόνα απόδοσης αρχικής τιμής σε πίνακα. Γράφουμε

```
char isbn[] = {'0','-','4','9','-','7','4','3','-','3','\0'};
```

Στην πράξη χρησιμοποιείται η εναλλακτική και πιο συμπαγής μορφή με χρήση αλφαριθμητικής σταθεράς:

```
char isbn[] = "0-49-743-3";
```

Θα πρέπει να προσεχθεί ότι στη δήλωση με λίστα στοιχείων ο προγραμματιστής πρέπει να περιλάβει ως τελευταία τιμή το null. Στον δεύτερο τρόπο απόδοσης αρχικής τιμής, αυτή την εργασία την εκτελεί αυτόματα ο μεταγλωττιστής.

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Ανάγνωση αλφαριθμητικού

Η εισαγωγή αλφαριθμητικού από την κύρια είσοδο γίνεται με την scanf() και με τον ίδιο προσδιοριστή που χρησιμοποιείται για την εκτύπωση. Η πρόταση

```
scanf( “%s”, isbn );
```

διαβάζει την κύρια είσοδο ως αλφαριθμητικό και αποθηκεύει την τιμή στη μεταβλητή isbn.

Δε χρειάζεται ο τελεστής & πριν από το όνομα της μεταβλητής isbn όπως συνέβαινε με τους άλλους τύπους δεδομένων, γιατί το όνομα του πίνακα αναπαριστά τη διεύθυνση του πρώτου στοιχείου του πίνακα.

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Εισαγωγή αλφαριθμητικού

- Εναλλακτικά, η εισαγωγή αλφαριθμητικού μπορεί να γίνει με χρήση της συνάρτησης gets(), η γενική μορφή της οποίας είναι

gets(όνομα_πίνακα_χαρακτήρων)

- Καλείται η gets() με το όνομα του πίνακα χαρακτήρων ως όρισμα, χωρίς δείκτη. Με την επιστροφή από τη gets() το αλφαριθμητικό θα έχει περασθεί στον πίνακα χαρακτήρων. Η gets() θα διαβάζει χαρακτήρες από το πληκτρολόγιο έως ότου πατηθεί το ENTER.

- Για να διαβασθεί ένα στοιχείο πίνακα αλφαριθμητικών χρησιμοποιούνται οι προτάσεις:

scanf("%s", stringArray[i]); και gets(stringArray[i]);

Πανεπιστήμιο Δυτικής Αττικής Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Παρατηρήσεις στην εισαγωγή αλφαριθμητικού

1. Η συνάρτηση **scanf()** διαβάζει εσφαλμένα αλφαριθμητικά που περιέχουν λευκούς χαρακτήρες, καθώς σταματά την ανάγνωση στην εμφάνιση του πρώτου λευκού χαρακτήρα. Αντίθετα η συνάρτηση **gets()** διαβάζει τους λευκούς χαρακτήρες.

2. Θα πρέπει να σημειωθεί ότι τόσο η **scanf()** όσο και η **gets()** δεν εκτελούν έλεγχο ορίων στον πίνακα χαρακτήρων με τον οποίο καλούνται. Εάν π.χ. δηλωθεί **char stringName[100]** και το αλφαριθμητικό είναι μεγαλύτερο από το μέγεθος του **stringName**, ο πίνακας θα ξεπερασθεί. Επαφίεται στον προγραμματιστή να φροντίζει ώστε να μην προκληθεί υπέρβαση ορίων.

3. Όταν χρησιμοποιούνται πίνακες αλφαριθμητικών, η πρόταση

```
scanf( "%c", stringArray[i][j] );
```

διαβάζει τον χαρακτήρα του αλφαριθμητικού **i+1**, ο οποίος βρίσκεται στη θέση **j+1**.

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Εκτύπωση αλφαριθμητικού

- Η εκτύπωση αλφαριθμητικής σταθεράς γίνεται με την printf() χωρίς τη χρήση προσδιοριστή. Απλώς της περνάμε την προς εκτύπωση αλφαριθμητική σταθερά:

```
printf( “Hello” );
```

- Η εκτύπωση αλφαριθμητικού γίνεται με την printf() χρησιμοποιώντας τον προσδιοριστή %s. Η παρακάτω πρόταση

```
printf( “The ISBN code is: %s”, isbn );
```

θα έχει ως αποτέλεσμα να τυπωθεί στην οθόνη η πρόταση

```
The ISBN code is: 0-49-743-3
```

Πανεπιστήμιο Δυτικής Αττικής | Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Εκτύπωση αλφαριθμητικού

•Εναλλακτικά, η εκτύπωση αλφαριθμητικής σταθεράς και αλφαριθμητικού μπορεί να γίνει με χρήση της συνάρτησης puts(), η γενική μορφή της οποίας είναι: puts(όνομα_πίνακα_χαρακτήρων)

•Καλείται η puts() με το όνομα του πίνακα χαρακτήρων ως όρισμα, χωρίς δείκτη, π.χ. puts(isbn). Βέβαια, η puts() παρουσιάζει το μειονέκτημα ότι δεν παρέχει δυνατότητες μορφοποίησης της εξόδου.

•Για να εκτυπωθεί ένα στοιχείο πίνακα αλφαριθμητικών χρησιμοποιούνται οι προτάσεις:

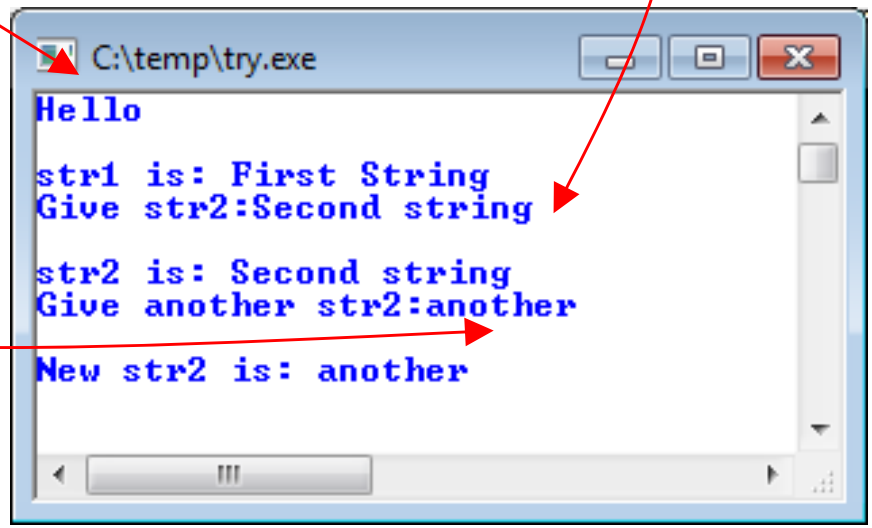
printf("%s", stringArray[i]); και puts(stringArray[i]);

ενώ η πρόταση printf("%c", stringArray[i][j]); εκτυπώνει τον χαρακτήρα του αλφαριθμητικού i+1, ο οποίος βρίσκεται στη θέση j+1.

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Παράδειγμα 5: Εισαγωγή και εκτύπωση αλφαριθμητικών

```
#include <stdio.h>
#define STA "Hello"
main()  {
        char str1[ ]= "First String";     char str2[81];
        puts(STA);
        printf( "\nstr1 is: %s\nGive str2:",str1 );
        gets(str2);
        printf( "\nstr2 is: %s\nGive another str2:",str2 );
        scanf( "%s",str2 );
        printf( "\nNew str2 is: " );
        puts(str2);
}
```



Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Συναρτήσεις αλφαριθμητικών
#include <string.h>

Λειτουργία	Όλοι οι χαρακτήρες	Οι *n* πρώτοι χαρακτήρες
Εύρεση μήκους αλφαριθμητικού	strlen()	
Αντιγραφή αλφαριθμητικού	stcpy()	strncpy()
Συνένωση δύο αλφαριθμητικών	strcat()	strncat()
Σύγκριση δύο αλφαριθμητικών	strcmp()	strncmp()
Εύρεση χαρακτήρα σε αλφαριθμητικό	strchr()	strrchr()
Εύρεση αλφαριθμητικού σε αλφαριθμητικό	strstr()	

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Παρατήρηση

Στις συναρτήσεις **stcpy()**και **strcat()** ελλοχεύει ο κίνδυνος να ξεπερασθούν τα όρια του πίνακα χαρακτήρων προορισμού, με αποτέλεσμα να δημιουργηθούν απροσδιόριστες καταστάσεις. Εάν π.χ. έχει ορισθεί ο πίνακας χαρακτήρων **char array[4]**, ο οποίος καταλαμβάνει τις θέσεις **1000** έως και **1003** στον χάρτη μνήμης του σχήματος της επόμενης διαφάνειας, και επιχειρηθεί να τοποθετηθεί σε αυτό το αλφαριθμητικό: **"character"** θα τεθεί θέμα απροσδιοριστίας.

Συγκεκριμένα, στις θέσεις **1004**-**1009**, τις οποίες δε διαχειρίζεται ο **array**, θα τοποθετηθούν οι χαρακτήρες '**a**', '**c**', '**t**', '**e**', '**r**', '**\0**'. Οι θέσεις όμως αυτές μπορεί να χρησιμοποιούνται από άλλες μεταβλητές και με την τροποποίηση του περιεχομένου τους οι νέες τιμές να προκαλέσουν σημασιολογικά σφάλματα στο πρόγραμμα.

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Παρατήρηση

Η εγγραφή δεδομένων εκτός των ορίων ενός πίνακα στην περιοχή προσωρινής αποθήκευσης ονομάζεται **υπερχείλιση της περιοχής προσωρινής αποθήκευσης** (buffer overflow). Για αυτόν τον λόγο θα πρέπει ο προγραμματιστής είτε να έχει προβλέψει επαρκή χώρο είτε σε κάθε χρήση αυτών των συναρτήσεων να εκτελεί έλεγχο ορίων πριν τις χρησιμοποιήσει.

	διεύθυνση	τιμή
array[0]	1000	c
array[1]	1001	h
array[2]	1002	a
array[3]	1003	r
array[4]	1004	a
array[5]	1005	c
array[6]	1006	t
array[7]	1007	e
array[8]	1008	r
array[9]	1009	\0

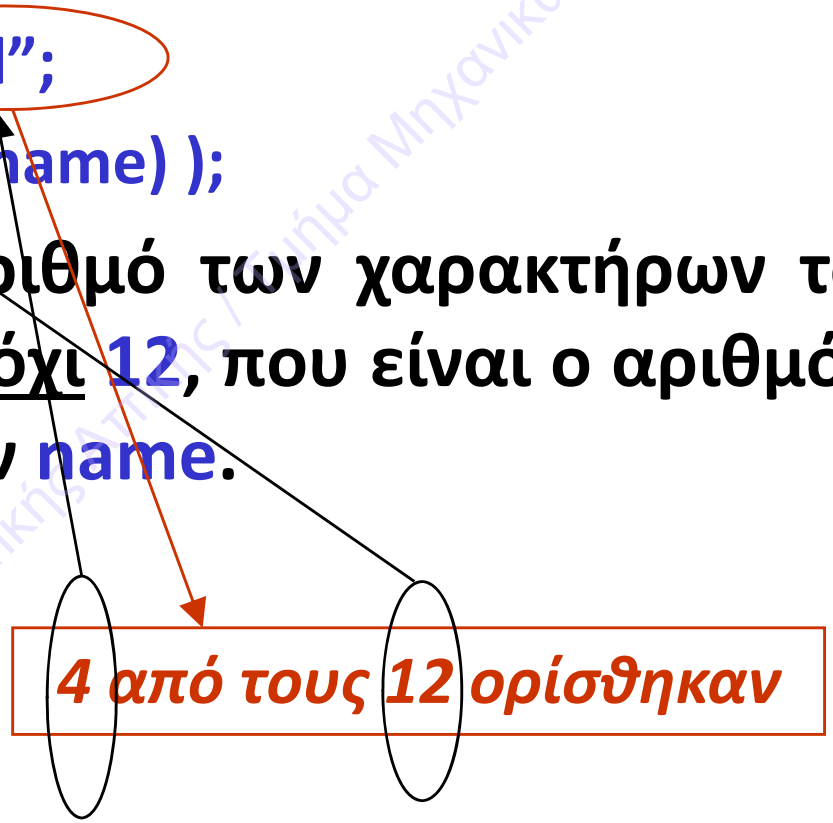
Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Η συνάρτηση μήκους αλφαριθμητικού (strlen)

Η συνάρτηση επιστρέφει τον αριθμό χαρακτήρων του αλφαριθμητικού, χωρίς να συμπεριλαμβάνει το μηδενικό χαρακτήρα. Το παρακάτω τμήμα κώδικα

```
char name[12] = “abcd”;
printf( “%d\n”, strlen(name) );
```

θα τυπώσει τον αριθμό των χαρακτήρων του αλφαριθμητικού name, δηλαδή 4 κι όχι 12, που είναι ο αριθμός των στοιχείων του πίνακα χαρακτήρων name.



Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Παράδειγμα 7:

```
#include <stdio.h>
#include <string.h>   /* για τη strlen() */

void main() {
   char msg[20] = {"Hello world!"};
   int cnt;
    cnt = strlen(msg);
    printf("length of\"%s\" is: %d\n",msg,cnt);
 }
```

Αποτέλεσμα:

> length of "Hello world!" is: 12

>

Σημείωση : ο χαρακτήρας null δεν προσμετράται

Πανεπιστήμιο Δυτικής Αττικής / Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών