ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά μαθήματα ΠΠ

Προγραμματισμός Η/Υ

Ενότητα 6: Πίνακες και Δείκτες

Νίκος Καρακαπιλίδης, Καθηγητής
Δημήτρης Σαραβάνος, Καθηγητής
Πολυτεχνική Σχολή
Τμήμα Μηχανολόγων & Αεροναυπηγών Μηχανικών

Σκοποί ενότητας

- Κατανόηση της έννοιας του πίνακα
- Κατανόηση της έννοιας του δείκτη

Περιεχόμενα ενότητας

- Ο τύπος του Πίνακα (βλ. ενότητα #2α)
- Ο τύπος του Δείκτη
- Περί αλφαριθμητικών δεδομένων

Μέρος 1ο

Ο τύπος του Πίνακα

Ο τύπος του Πίνακα

Ορισμός και αρχικοποίηση

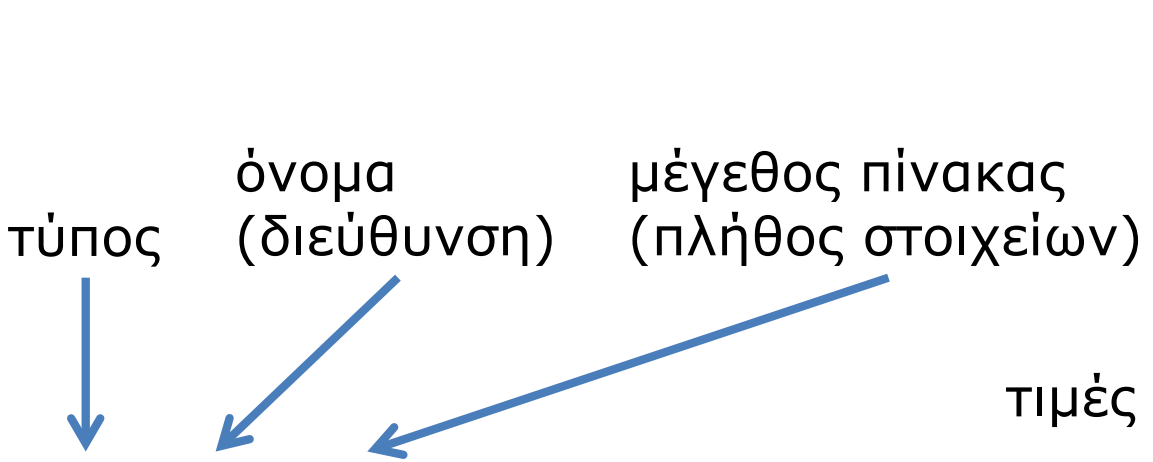
Τύπος πίνακα

Συλλογή μεταβλητών του ίδιου τύπου, οι οποίες είναι **αποθηκευμένες σε διαδοχικές θέσεις μνήμης**

- Είναι από τις πιο βασικές δομές δεδομένων (data structures)
- Πετυχαίνεται γρήγορη πρόσβαση και προτιμούνται από άλλες δομές δεδομένων

Τύπος πίνακα

Παράδειγμα: μέσες μηνιαίες θερμοκρασίες ενός έτους




```
float temp[12] =
{10,12,13,17,20,25,30,31,30,27,20,17};
```

10	temp [0]
12	temp [1]
13	temp [2]
17	.
20	.
25	.
…	.
27	.
20	temp [10]
17	temp [11]

Προσοχή στην αναφορά των στοιχείων ενός πίνακα

Τύπος πίνακα

Απόδοση αρχικής τιμής

- Αρχικοποίηση όλων των στοιχείων

```
float grade[5] = {6.5,5,10,8,3.5};
```

- Μερική αρχικοποίηση

```
float grade[5] = {6.5,5,10};
/* αρχικοποιεί τα 3 πρώτα στοιχεία,
   αφήνοντας τα άλλα απροσδιόριστα */
```

Τύπος πίνακα

- Παράλειψη ορισμού του μεγέθους του πίνακα

  float grade[] = {6.5,5,10,8,3.5};

- Αρχικό (πρώτο) στοιχείο πίνακα

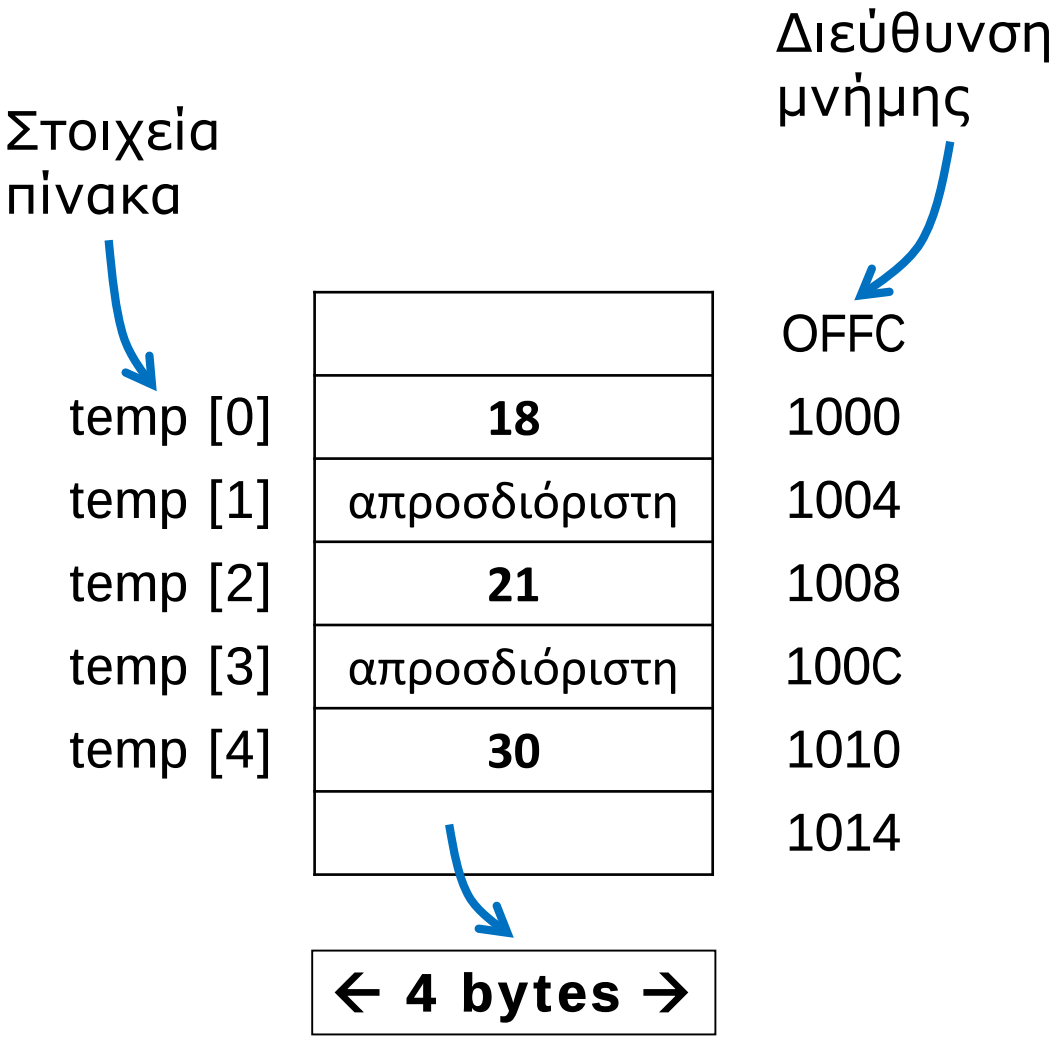
  grade[0] → 6.5

Προσοχή: αναφερόμαστε στα στοιχεία ενός πίνακα με τον αριθμό που δηλώνει το δείκτη τους, π.χ. grade[3]

Αποθήκευση πινάκων στη μνήμη

Παράδειγμα

```
int temp[5];
...
temp[0]=18;
temp[2]=21;
...
temp[4]=temp[0]+12;
```



Τύπος πίνακα

Η επεξεργασία ενός πίνακα γίνεται ανά στοιχείο

Παραδείγματα:

```
printf("%d", x[0]);
x[3] = 8258;
sum = x[0] + x[1];
sum += x[2];
x[3] += 1;
x[2] = x[0] + x[1];
```

Τύπος πίνακα

Αναφορά σε στοιχεία πίνακα (δείκτες πίνακα)

Παραδείγματα:

```
x[i] = 0;
x[i] = x[j];
x[j+k*4]= x[u] + 3;
x[x[i]]= p;
diff = x[y]-x[foo()];
total += x[i++];
```

Προσοχή: πρόσβαση σε μη υπαρκτό στοιχείο ενός πίνακα μπορεί να οδηγήσει είτε στον τερματισμό του προγράμματος είτε (το πιθανότερο) σε λανθασμένα αποτελέσματα

Αρχικοποίηση πίνακα

Παραδείγματα

```
/* αρχικοποίηση όλων των
   στοιχείων του πίνακα x σε 0 */

#define MAX  100
...
int x[MAX], i;
for(i=0;i<MAX;++i)
        x[i]=0;
...
```

```
/* αρχικοποίηση στοιχείων πίνακα
   μέσω συνάρτησης */

#define MAX  100
void init_table(int x[], int size)
        {
        int i;
        for(i=0;i<size;++i)
                x[i]=0;
        }

int main()
{
        int table[MAX];
        init_table(table,MAX);
        ....
}
```

Παράλληλοι πίνακες

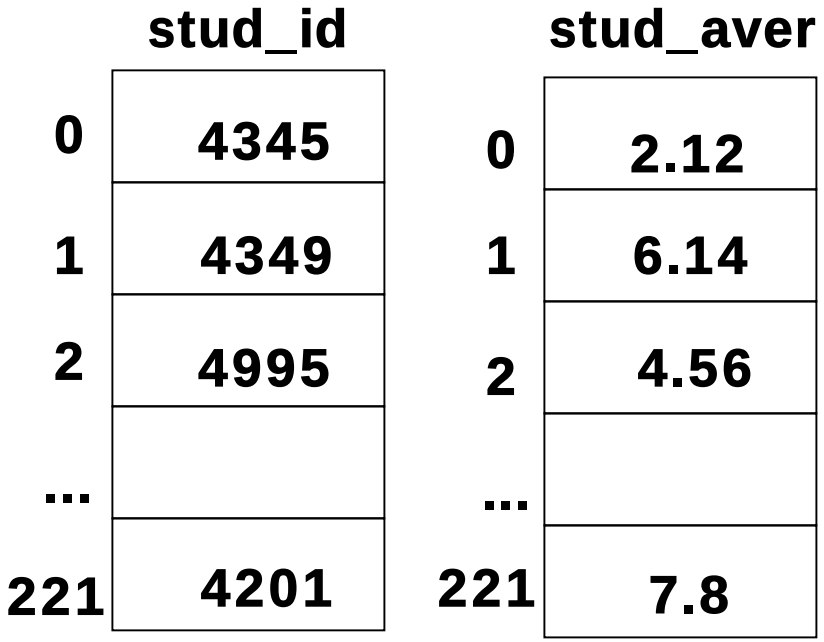
Περιέχουν συγγενείς πληροφορίες για το ίδιο σύνολο οντοτήτων

Παράδειγμα:

```
#define NUM_STUD 222
int stud_id [NUM_STUD];
float stud_aver [NUM_STUD];
```

Παράλληλοι πίνακες

Ο πίνακας **stud_id** περιέχει τους αριθμούς μητρώου των πρωτοετών φοιτητών του ακαδημαϊκού έτους 2009-10, ενώ ο πίνακας **stud_aver** τους μέσους όρους από τη βαθμολογία των ασκήσεων του μαθήματος, για την ίδια ομάδα φοιτητών



Παράλληλοι πίνακες

```
#define NUM_STUD 222
...
void display_id_grade(int table_id[], float table_grade[], int size)
{
    int j;
    for(j=0;j<size;++j)
        printf("Student with id: %d, grade: %f\n",
table_id[j],table_grade[j]);
}
...
main
{
...
int stud_id [NUM_STUD];
float stud_aver [NUM_STUD];
...
display_id_grade(stud_id[], stud_aver[], NUM_STUD)
...
}
```

Πολυδιάστατοι πίνακες

Πίνακες των οποίων κάθε στοιχείο είναι πίνακας

Παράδειγμα: πίνακας βαθμολογίας με 222 στοιχεία (ένα για κάθε φοιτητή του Α’ έτους), όπου κάθε ένα από αυτά είναι ένας πίνακας 6 στοιχείων (ένα για κάθε μάθημα του συγκεκριμένου φοιτητή αυτό το εξάμηνο)

```
float grade[222][6];
```

Πολυδιάστατοι πίνακες

Που αναφέρονται οι παρακάτω εκφράσεις;

grade[0][5];

grade[1][0];

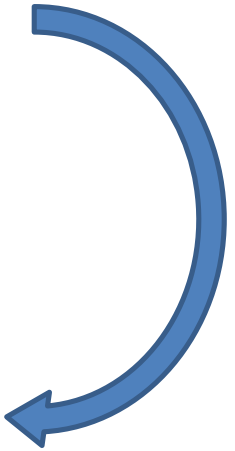
grade[199][3];

Προσοχή:
Προτάσεις της μορφής **grade[0,5]** (οι οποίες υπάρχουν σε γλώσσες όπως η Pascal και η Fortran) – αντί του ορθού **grade[0][5]**- οδηγούν συνήθως σε ανεπιθύμητα αποτελέσματα

Πολυδιάστατοι πίνακες

Αποθήκευση

17	24	6
5	16	4
12	11	6



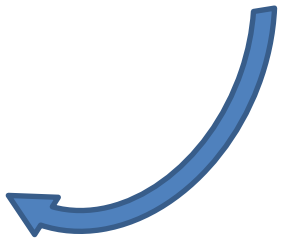
```
static int example[3][3] = {     {17,24,6},
                                 {5,16,4},
                                 {12,11,6}
                           };
```

Πολυδιάστατοι πίνακες

Αρχικοποίηση

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

1	2	3
4	0	0
5	6	7
0	0	0
0	0	0



Μέρος 2ο

Ο τύπος του Δείκτη

Ο τύπος του Δείκτη

Η έννοια του Δείκτη

Ο τύπος του Δείκτη

Από τις δυσκολότερες έννοιες της C

- Η χρήση τους αποτελεί αιτία δυσκολιών στον εντοπισμό σφαλμάτων

Η έννοια του δείκτη

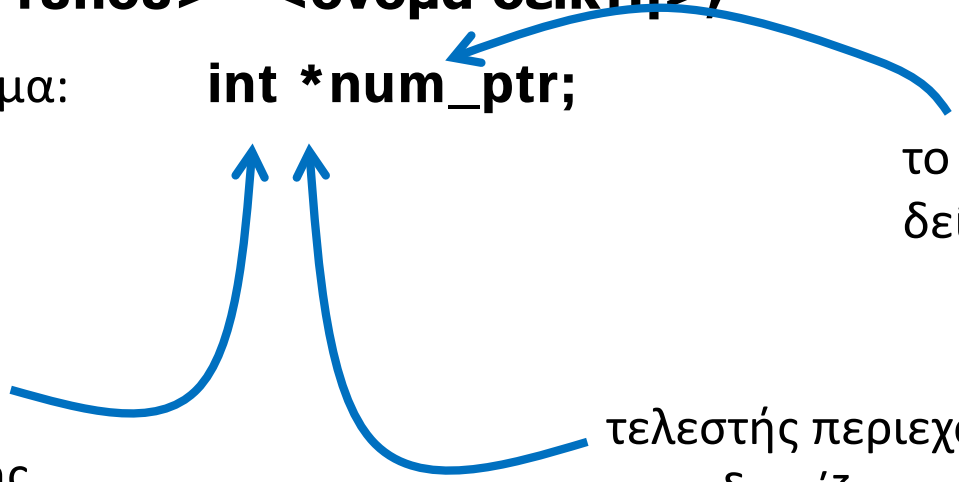
- Κάθε μεταβλητή σχετίζεται με μια θέση στην κύρια μνήμη του Η/Υ (κάθε τέτοια θέση έχει τη δικιά της, ξεχωριστή διεύθυνση)
- Ο δείκτης δεν είναι τίποτε παραπάνω από μια μεταβλητή η οποία χρησιμοποιείται για την **αποθήκευση μιας διεύθυνσης** της κύριας μνήμης του Η/Υ

Ο τύπος του Δείκτη

Δήλωση δείκτη

<όνομα τύπου> *<όνομα δείκτη>;

Παράδειγμα: **int *num_ptr;**



το όνομα του δείκτη

τύπος του στοιχείου που δείχνει ο δείκτης

τελεστής περιεχομένου - προσδιορίζει τη **num_ptr** ως (μεταβλητή) δείκτη

Ο τύπος του Δείκτη

Αρχικοποίηση δείκτη

Ένας δείκτης πρέπει να δείχνει σε μια θέση μνήμης που ανήκει στο πρόγραμμα

Αρχικοποίηση με άμεση διεύθυνση

- Πολύ «επικίνδυνες», πρέπει να αποφεύγονται

```
num_ptr = 1000;  /* int *num_ptr = 1000; */
```

Αρχικοποίηση με διεύθυνση μεταβλητής ακεραίου τύπου (χρήση του τελεστή διεύθυνσης **&**)

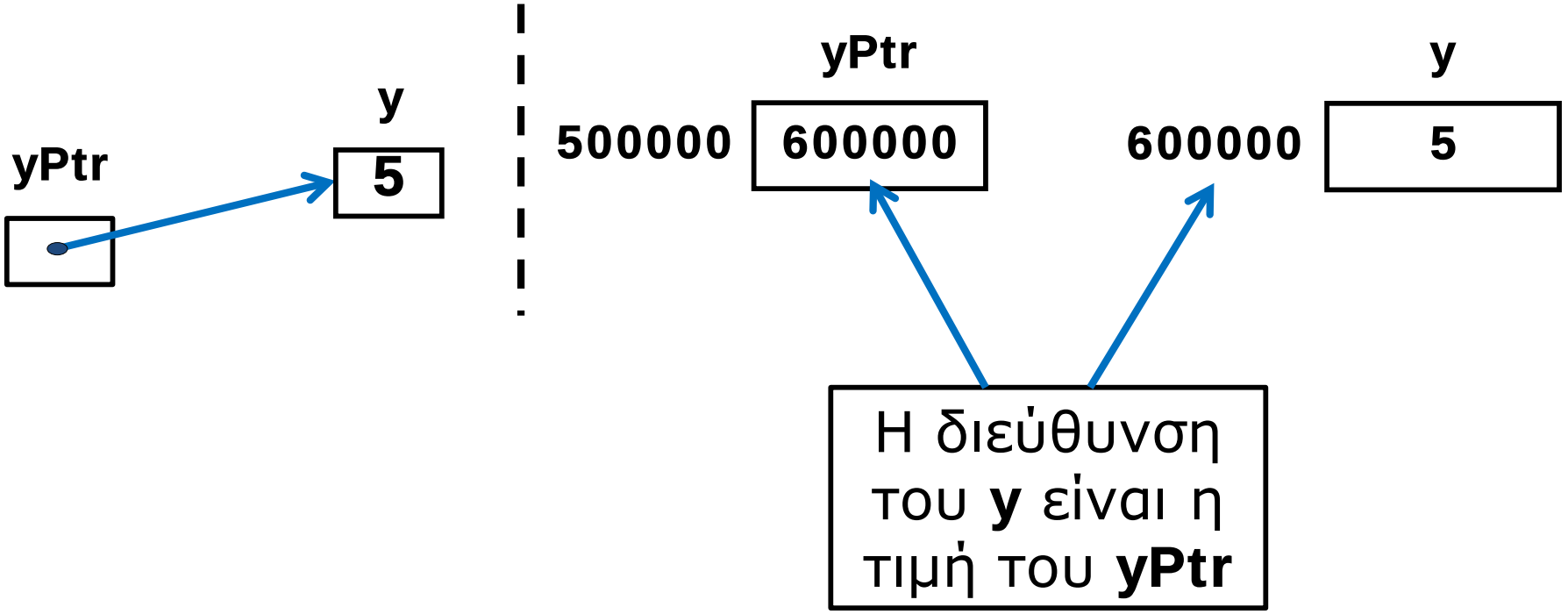
```
num_ptr = &num;  /* int *num_ptr = &num; */
```

Ο τύπος του Δείκτη

Παράδειγμα (χρήση τελεστή διεύθυνσης **&**)

```
int y = 5;
    int *yPtr;
    yPtr = &y; /* η yPtr παίρνει τη διεύθυνση του y */
               /* η yPtr δείχνει στην ("points to") y */
```

Ο τύπος του Δείκτη



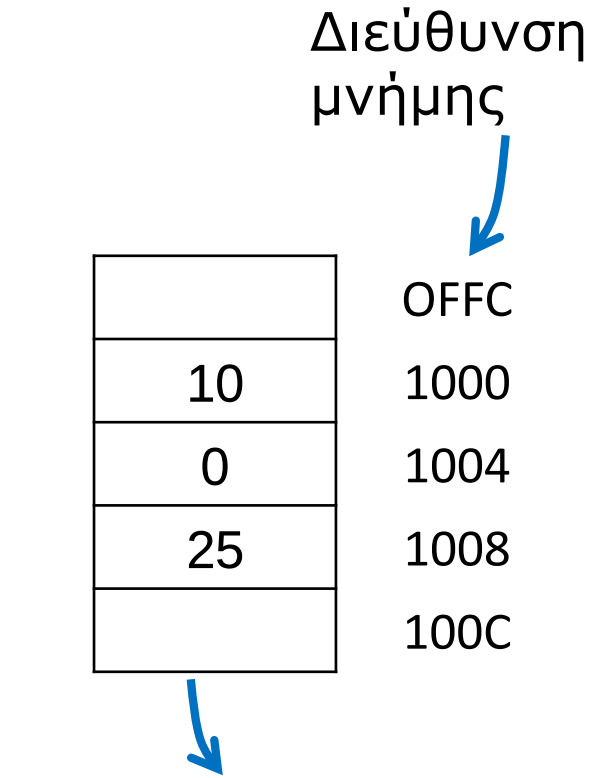
Προσοχή:
Η ***yPtr** επιστρέφει **y** (επειδή η **yPtr** δείχνει στην **y**)

Η πρόταση
***yPtr = 7;**
αλλάζει την τιμή του **y** σε 7

Ο τύπος του Δείκτη

Παράδειγμα

```
int num=10;
int count=0;
int mcount=25;
...
printf("%d\t %p\n", num, &num);
printf("%p\t %p\n", &count, &num);
...
```




```
10      1000
1004    1000
```

Output προγράμματος

Μέρος 3ο

Περί αλφαριθμητικών δεδομένων

Περί αλφαριθμητικών δεδομένων

Συναρτήσεις αλφαριθμητικών

Αλφαριθμητικά δεδομένα

Ακολουθίες χαρακτήρων

- Ο ISBN κωδικός του βιβλίου σας είναι «960-8050-79-0»
- Η διεύθυνση ενός φοιτητή είναι «Γούναρη 234»

Αποθήκευση και διαχείριση αλφαριθμητικών στη C

- Μέσω πινάκων
  - Πίνακες χαρακτήρων που τερματίζουν με το μηδενικό (null) χαρακτήρα
  - Ο μεταγλωττιστής θέτει αυτόματα στο τέλος του αλφαριθμητικού ένα μηδενικό χαρακτήρα για να προσδιορίσει το τέλος του
  - Ο μηδενικός χαρακτήρας έχει ASCII κωδικό 0 και αναπαρίσταται με την ακολουθία διαφυγής **'\0'**
- Παραδείγματα δηλώσεων

```
char isbn[15], stud_address[30];
```

Αλφαριθμητικά δεδομένα

Αποθήκευση

‘H’	**‘e’**	**‘l’**	**‘l’**	**‘o’**	**‘\0’**

Αρχικοποίηση

```
char isbn[]=“960-8050-79-0”;
```

Αλφαριθμητικά δεδομένα

Εκτύπωση

Αλφαριθμητική σταθερά

```
printf("Hello");
printf("Ο ISBN κωδικός είναι: %s\n", isbn);
```

Αλφαριθμητικό

Εισαγωγή

```
scanf("%s",isbn);
```

Χωρίς τελεστή & πριν το όνομα της μεταβλητής

Ένα απλό παράδειγμα

Ανάγνωση αλφαριθμητικού από την είσοδο και εκτύπωσή του

```
#include<stdio.h>
#define MAX_CHAR 80

main()
{
  char str[MAX_CHAR];
  int i;

  printf("Enter a string: ");
  scanf("%s", str);  /* εναλλακτικά, scanf("%s", &str[0]); */

  for (i=0;i<10;i++)
    printf("%s\n", str);

  exit(0);
}
```

Συναρτήσεις χειρισμού αλφαρ/κών

Συνάρτηση μήκους αλφαριθμητικού

strlen()

Επιστρέφει τον αριθμό χαρακτήρων του αλφαριθμητικού (χωρίς να συμπεριλαμβάνει το μηδενικό χαρακτήρα)

Παράδειγμα

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

Output προγράμματος → **4**

Συναρτήσεις χειρισμού αλφαρ/κών

Συνάρτηση αντιγραφής αλφαριθμητικών

strcpy()

Αντιγράφει ένα αλφαριθμητικό από έναν πίνακα σε έναν άλλο

- Δέχεται δύο ορίσματα, το όνομα του πίνακα προορισμού (πρώτο όρισμα) και το όνομα του πίνακα πηγής (δεύτερο όρισμα)

Συναρτήσεις χειρισμού αλφαρ/κών

Παράδειγμα

```
char name_1[12]=“abcd”;
char name_2[12]=“ef”;
strcpy(name_1,name_2);
printf(“%s\n”, name_1);
```

Output προγράμματος → **ef**

Συναρτήσεις χειρισμού αλφαρ/κών

Συνάρτηση συνένωσης αλφαριθμητικών

strcat()

Δέχεται δύο ορίσματα, τα οποία συνενώνει

- «Προσθέτει» στο τέλος του αλφαριθμητικού που προσδιορίζεται από το πρώτο όρισμα τα στοιχεία του αλφαριθμητικού που προσδιορίζεται από το δεύτερο όρισμα

Συναρτήσεις χειρισμού αλφαρ/κών

Παράδειγμα

Προσοχή:
Ο πίνακας **name_1** πρέπει να έχει κατάλληλο «μέγεθος», έτσι ώστε να χωράνε και τα στοιχεία του πίνακα **name_2**

```
char name_1[12]="abcd";
char name_2[12]="ef";
strcat(name_1,name_2);
printf("%s\n", name_1);
```

Output προγράμματος → **abcdef**

Συναρτήσεις εισόδου/εξόδου χαρακτήρων

int getchar()

- διαβάζει τον επόμενο χαρακτήρα από την είσοδο (και κινεί τον δρομέα στον επόμενο χαρακτήρα)

void putchar(int)

- τυπώνει χαρακτήρα στη μονάδα εξόδου

Συναρτήσεις Ε/Ε χαρακτήρων

Παραδείγματα χρήσης

Ανάγνωση σειράς χαρακτήρων απροσδιόριστου μεγέθους

- Αλγόριθμος;

Υπολογισμός μεγέθους σειράς χαρακτήρων

- Αλγόριθμος;

Υπολογισμός εμφανίσεων χαρακτήρων

- Αλγόριθμος;

Συναρτήσεις Ε/Ε χαρακτήρων

Ανάγνωση σειράς χαρακτήρων

```
int c;
c = getchar();          /* διάβασε τον πρώτο χαρακτήρα */
while(c != EOF){        /* όχι τέλος του αρχείου */

  c = getchar();        /* διάβασε τον επόμενο χαρακτήρα */

}
```

Συναρτήσεις Ε/Ε χαρακτήρων

Echo σειράς χαρακτήρων

```
int c;
c = getchar();          /* διάβασε τον πρώτο χαρακτήρα */
while(c != EOF){        /* όχι τέλος του αρχείου */

  putchar(c);           /* τύπωσε χαρακτήρα */
  c = getchar();        /* διάβασε τον επόμενο χαρακτήρα */

}
```

Συναρτήσεις Ε/Ε χαρακτήρων

Υπολογισμός μεγέθους σειράς χαρακτήρων

```
int c;
int size=0;                 /* αρχικοποίηση */

c = getchar();              /* διάβασε τον πρώτο χαρακτήρα */

while(c != EOF){            /* όχι τέλος του αρχείου */
  size=size+1;              /* «υπολόγισε» άλλον ένα χαρακτήρα */
  c = getchar();            /* διάβασε τον επόμενο χαρακτήρα */
}
```

Τέλος Ενότητας

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου των διδασκόντων.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Πατρών**» έχει χρηματοδοτήσει μόνο την 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.

Σημείωμα Αναφοράς

Σημείωμα Αδειοδότησης

Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

Σημείωμα Χρήσης Έργων Τρίτων