



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά
μαθήματα ΠΠ

Προγραμματισμός Η/Υ

Ενότητα 2α: Εισαγωγή στη C (Μέρος Πρώτο)

Νίκος Καρακαπιλίδης, Καθηγητής

Δημήτρης Σαραβάνος, Καθηγητής

Πολυτεχνική Σχολή

Τμήμα Μηχανολόγων & Αεροναυπηγών Μηχανικών

Σκοποί ενότητας

- Αναφορά στις λέξεις κλειδιά που χρησιμοποιούνται στη C και πως αυτές συντάσσονται μέσα στο πρόγραμμα
- Κατανόηση της έννοιας της μεταβλητής



Περιεχόμενα ενότητας

- Στοιχεία της Γλώσσας Προγραμματισμού C
- Οι έννοιες της Μεταβλητής και της Σταθεράς
- Τύποι Δεδομένων στη C
- Απλά προγράμματα στη C



Μέρος 1^ο

Στοιχεία της Γλώσσας Προγραμματισμού C

Στοιχεία της Γλώσσας Προγραμματισμού C

Αλφάβητο, Συντακτικό Γλώσσας, Λεξιλόγιο

Αλφάβητο

Χρήση χαρακτήρων αλφαβήτου για σχηματισμό λέξεων, συμβόλων και εντολών

ANSI πρότυπο → 96 χαρακτήρες

- Χαρακτήρας κενού, χαρακτήρες ελέγχου οριζόντιου και κάθετου στηλοθέτη, αλλαγής σελίδας, νέας γραμμής (5)
- a b c ... x y z (26)
- A B C ... X Y Z (26)
- 0 1 2 3 ... 9 (10)
- _ { } [] # () < > % : ; . ? * + - / ^ & | ~ ! = , \ " ' (29)

Μέρος ευρύτερου συνόλου χαρακτήρων

- ASCII (American Standard Code for Information Interchange)



Συντακτικό Γλώσσας

Οι συντακτικοί κανόνες ορίζουν τον τρόπο με τον οποίο οι λέξεις (και τα σύμβολα) μιας γλώσσας προγραμματισμού μπορούν να συγκροτήσουν προτάσεις και να δημιουργήσουν προγράμματα

Εναλλακτικοί τρόποι έκφρασης του συντακτικού μιας γλώσσας

- Backus-Naur-Form (BNF)
- Συντακτικό διάγραμμα (syntax graph)
- Γραμματικές χωρίς συμφραζόμενα (context-free grammars)
 - **<stmt> --> if (<expr>) <stmt> else <stmt>**

Συντακτικό Γλώσσας είναι ένα σύνολο κανόνων που καθορίζει αν μια πρόταση είναι ορθά ή όχι δομημένη



Λεξιλόγιο

Δεσμευμένες λέξεις (reserved words)

- Λέξεις κλειδιά (keywords)
- Ονόματα συναρτήσεων πρότυπης βιβλιοθήκης
 - **printf, abs, isdigit, ...**
- Ονόματα που περιέχονται σε αρχεία (επικεφαλίδας) της βασικής βιβλιοθήκης για ορισμό μακροεντολών (macro names)
 - **EOF, INT_MAX, ...**
- Ονόματα τύπων (type names) που ορίζει η βασική βιβλιοθήκη σε αρχεία επικεφαλίδας
 - **time_t, va_list, ...**



Λεξιλόγιο

- Ονόματα εντολών προ-επεξεργαστή (preprocessor)
 - **include, define, ...**
- Ονόματα που αρχίζουν με τον χαρακτήρα υπογράμμισης _
 - **_DATE_, _FILE_, ...**

Τελεστές (operators) είναι σύμβολα ή λέξεις που αναπαριστούν συγκεκριμένες διεργασίες (π.χ. ο τελεστής * αναπαριστά τη διεργασία του πολλαπλασιασμού)

Αναγνωριστές (identifiers) είναι λέξεις που κατασκευάζει ο προγραμματιστής



Λέξεις κλειδιά

Πίνακας 1: ANSI C

auto	double	int	struct
Break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while



Λέξεις κλειδιά

Περιορισμός για μια γλώσσα, αλλά αυξάνουν την αναγνωσιμότητα και αξιοπιστία των προγραμμάτων και επιταχύνουν τη διαδικασία της μεταγλώττισης

Πολλές είναι αποδεκτές σε πολλές γλώσσες προγραμματισμού



Αναγνωριστές (identifiers)

Λέξεις (ονόματα) που δίνει ο προγραμματιστής σε δικές του κατασκευές (μεταβλητές, σταθερές, συναρτήσεις και τύποι δεδομένων)

Ένα όνομα προσδιορίζει μοναδικά την κατασκευή στην οποία αποδόθηκε

Κανόνες για δημιουργία ενός ονόματος

- Πρέπει να αρχίζει υποχρεωτικά από γράμμα ή χαρακτήρα υπογράμμισης (`_`), ενώ μπορεί να περιέχουν γράμματα, αριθμούς και χαρακτήρες υπογράμμισης
- Δεν πρέπει να είναι ίδιο με μια δεσμευμένη λέξη
- Μικροί και κεφαλαίοι χαρακτήρες (του ίδιου γράμματος) θεωρούνται διαφορετικοί στη C (case sensitive)
- Η C δεν βάζει όριο στο μήκος ενός ονόματος (ο κάθε μεταγλωττιστής όμως βάζει)



Μέρος 2^ο

Οι έννοιες της Μεταβλητής και της
Σταθεράς

Οι έννοιες της Μεταβλητής και της Σταθεράς

Ορισμοί

Η έννοια της μεταβλητής

Θέματα διαχείρισης δεδομένων

- Αποθήκευση, πρόσβαση και τροποποίηση δεδομένων
- Χρήση ακριβούς διεύθυνσης μνήμης → καλή λύση;
- Χρήση συμβολικών ονομάτων → «μεταβλητές»

Το **όνομα της μεταβλητής** είναι άμεσα συνδεδεμένο με την **ακριβή διεύθυνση της μνήμης** όπου είναι αποθηκευμένη η **τρέχουσα τιμή της μεταβλητής**

- Αναφερόμαστε έτσι στην τιμή της χωρίς να χρειάζεται να γνωρίζουμε την ακριβή διεύθυνση της μνήμης όπου αυτή είναι αποθηκευμένη
- Επίσης, αναφερόμαστε στη θέση μνήμης (μέσω του ονόματος) για να τροποποιήσουμε την τρέχουσα τιμή της

Το όνομα της μεταβλητής έχει διπλή σημασία

count = count + 1;

Η έννοια της μεταβλητής

Horowitz (1995): Η μεταβλητή είναι μια τετράδα που αποτελείται από:

- Ένα όνομα
- Ένα σύνολο ιδιοτήτων
 - Είδος των τιμών που μπορεί να πάρει, χώρος μνήμης που απαιτεί για την αποθήκευσή της, μορφές επεξεργασίας που μπορεί να υποστεί κλπ.
 - Παράβαλε: **ονοματεπώνυμο** ενός φοιτητή και **βαθμό** που πήρε σε κάποιο μάθημα
 - Στη C, ο προσδιορισμός των ιδιοτήτων μιας μεταβλητής (**δήλωση μεταβλητής**) γίνεται πριν τη χρήση της
- Μια αναφορά
- Μια τιμή

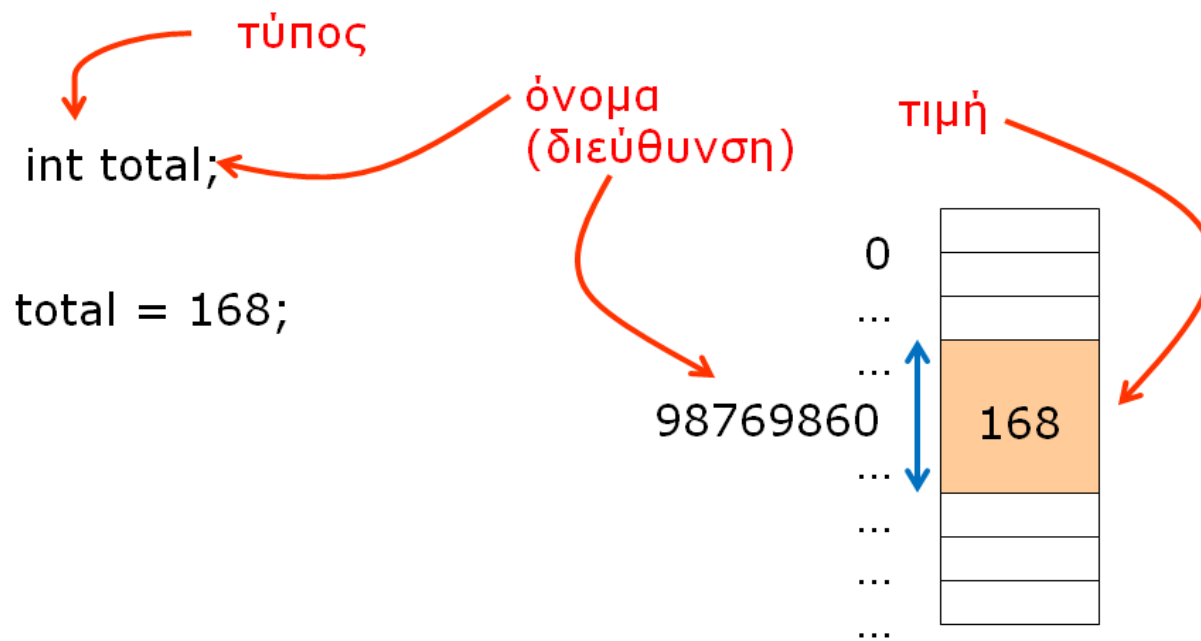


Η έννοια της μεταβλητής

Παραδείγματα

```
count = count + 1;
```

```
area = width * height;
```

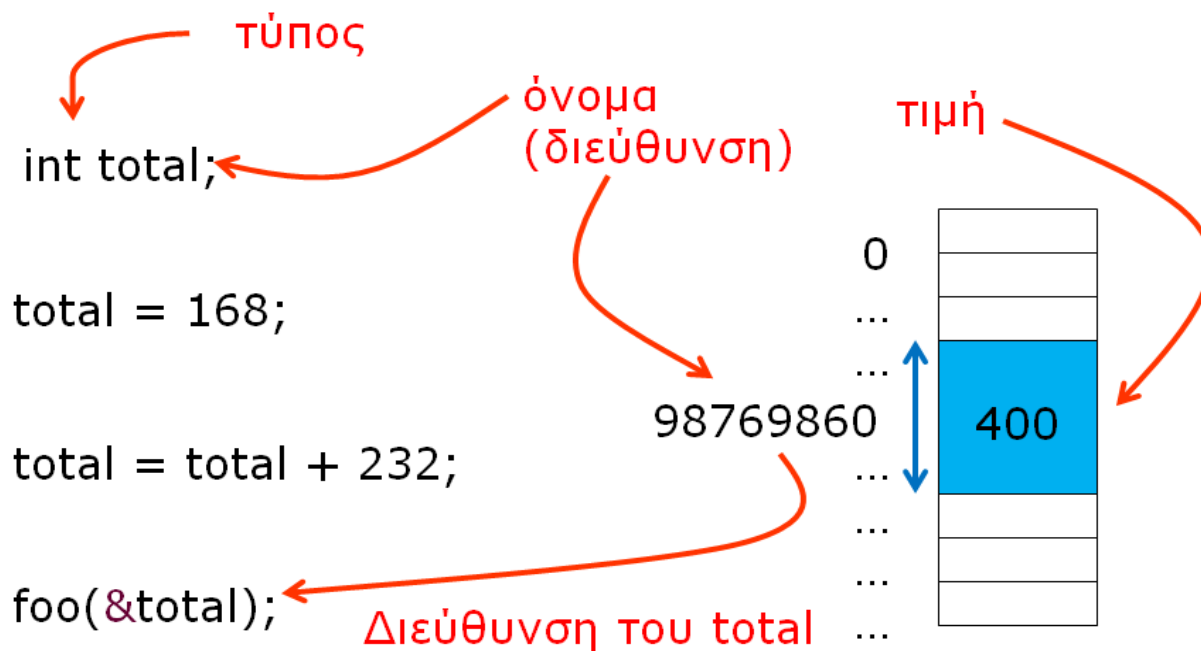


Η έννοια της μεταβλητής

Παραδείγματα

```
count = count + 1;
```

```
area = width * height;
```



Δήλωση μεταβλητής

Έχει ως αποτέλεσμα τη σύνδεση του ονόματος της μεταβλητής με:

- τον ανάλογο τύπο δεδομένων → γίνεται κατά τη μεταγλώττιση (compile time) του προγράμματος
- μια θέση μνήμης κατάλληλου μεγέθους → γίνεται κατά την εκτέλεση (run time) του προγράμματος



Δήλωση μεταβλητής

Μορφή δήλωσης

<όνομα τύπου δεδομένων> <λίστα ονομάτων μεταβλητών>;

```
int age;
```

```
int number_of_kids;
```

```
int age, number_of_kids;
```

<όνομα τύπου δεδομένων> <όνομα μεταβλητής>=<αρχική τιμή μεταβλητής>;

```
int age=35;
```



Μέρος 3^ο

Τύποι Δεδομένων στη C

Τύποι Δεδομένων στη C

Τύπος χαρακτήρα, ακεραίου, πραγματικών,
πίνακα

Τύποι Δεδομένων

Ένα σύνολο από αντικείμενα με κοινά χαρακτηριστικά

- Τα χαρακτηριστικά αυτά συνήθως εκφράζονται από το σύνολο των επιτρεπομένων πράξεων πάνω στα αντικείμενα αυτά

Ενσωματωμένοι και παραγόμενοι τύποι δεδομένων

- Ενσωματωμένοι → χαρακτήρας, ακέραιος, ...
- Παραγόμενοι → έστω η δομή φοιτητής (**struct student**) που ορίζει και ομαδοποιεί τις ιδιότητες των μεταβλητών που είναι απαραίτητες για τη διαχείριση των στοιχείων ενός φοιτητή

Βαθμωτοί (scalar) και συναθροιστικοί (aggregate) τύποι δεδομένων

- Βαθμωτοί → τύποι των οποίων οι τιμές βρίσκονται κατά μήκος μιας γραμμικής κλίμακας, π.χ. ακέραιος, χαρακτήρας, λογικός, ...
- Συναθροιστικοί → δημιουργούνται με συνδυασμό ενός ή περισσότερων βαθμωτών τύπων



Η έννοια της σταθεράς (const)

Οι σταθερές έχουν τιμή και όνομα, **αλλά όχι διεύθυνση**

- Ο μεταγλωττιστής αντικαθιστά το όνομα με την τιμή

Διευκολύνει τροποποιήσεις και την αναγνωσιμότητα ενός προγράμματος

- Παράβαλε: **PI** και **3.1453**

Παραδείγματα

```
#define PI 3.1453
```

```
#define YES 1
```

```
#define NO 1
```

```
#define TRUE 1
```

```
#define PISQUARE PI*PI
```

```
#define ERROR_SPACE_MESSAGE "error:run out of space"
```


Τύπος χαρακτήρα (char)

Αναπαράσταση απλών χαρακτήρων του αλφαβήτου της γλώσσας

- Εμφανίζεται στον πηγαίο κώδικα ανάμεσα σε απλά εισαγωγικά, π.χ. **'C'** , **'5'** , **'*'** , **'('** , ...

Παράδειγμα δήλωσης

```
char status = 'A';
```

Παράδειγμα εκτύπωσης

- Χρήση της **printf** μαζί με τον προσδιοριστή **%c**
printf("η κατάσταση είναι %c\n", status);
- Χρήση του προσδιοριστή **%d** → εμφάνιση του ASCII κωδικού του χαρακτήρα

```
printf("ο ASCII κώδικας του %c είναι %d\n", status, status);
```

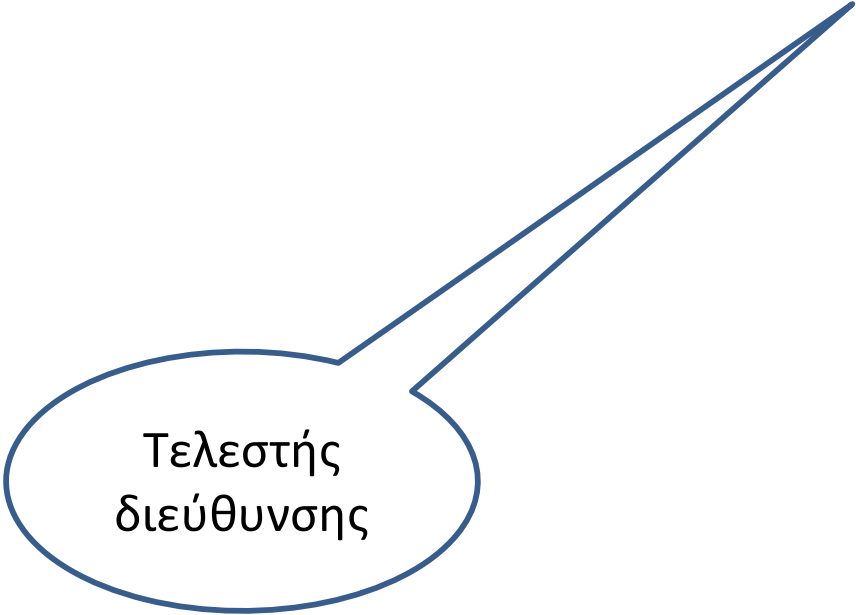
εμφάνιση την οθόνη → **ο ASCII κώδικας του A είναι 65**



Τύπος χαρακτήρα (char)

Παράδειγμα εισαγωγής

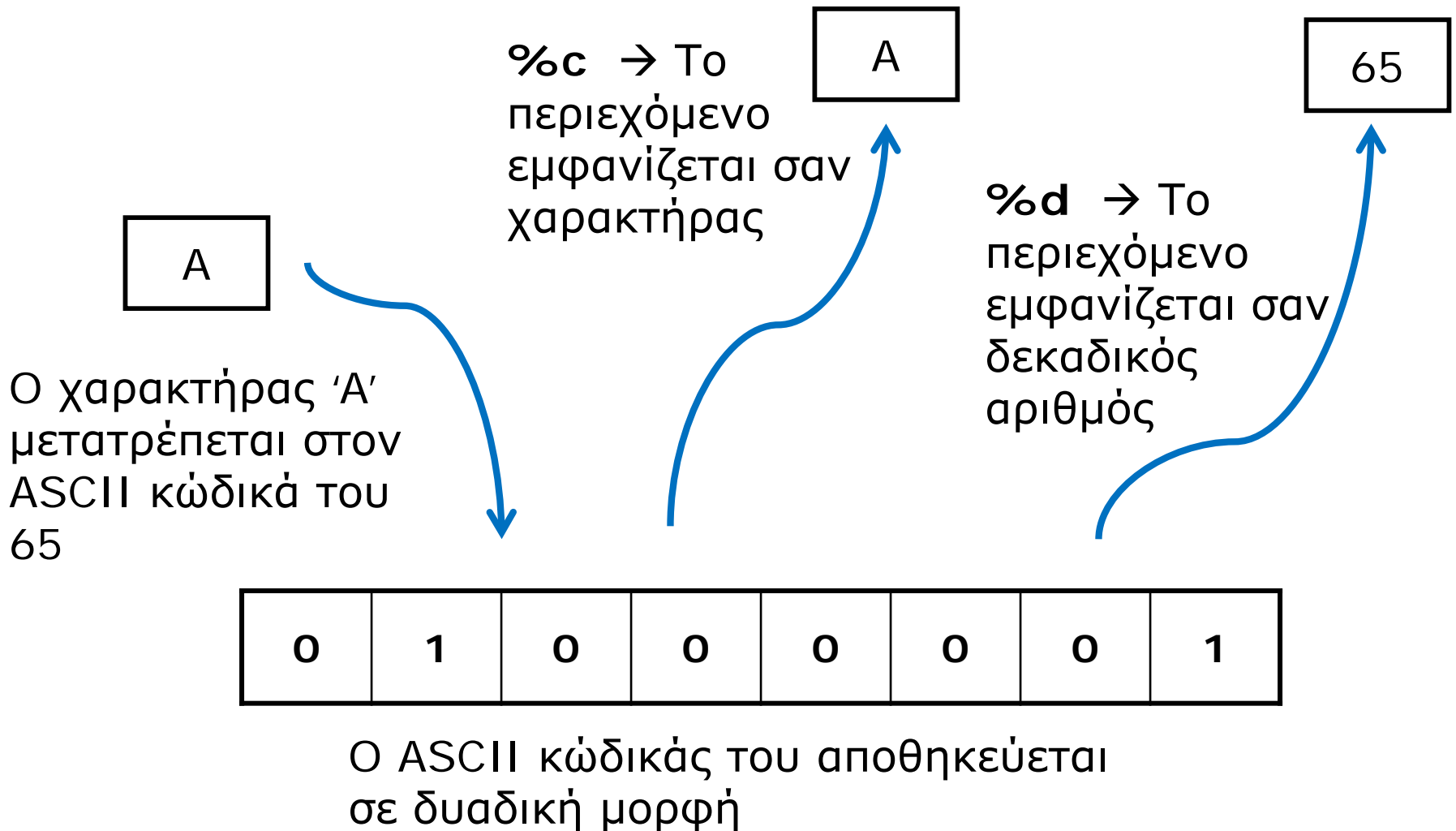
- Χρήση της **scanf** μαζί με τον προσδιοριστή **%c** και τον χαρακτήρα **&**
scanf("%c",&char1);



Τελεστής
διεύθυνσης



Τύπος char - Αποθήκευση στη μνήμη



Ένα απλό πρόγραμμα

```
1  /* πρόγραμμα ανάγνωσης ενός χαρακτήρα - τυπώνει τον χαρακτήρα και
2  τον επόμενο του και τους ASCII κωδικούς τους */
3  #include <stdio.h>
4
5  main()
6  {
7      char ch, next_ch;                /* δήλωση μεταβλητών */
8
9      printf("Δώσε ένα χαρακτήρα:\t"); /* prompt */
10     scanf( "%c", &ch );              /* read a character */
11     printf( "Ο ASCII κώδικας του χαρακτήρα %c είναι %d\n", ch, ch );
12     next_ch = ch + 1;                /* βρίσκει τον επόμενο χαρακτήρα */
13     printf( "Ο ASCII κώδικας του χαρακτήρα %c είναι %d\n", next_ch, next_ch );
14
15 }
```

Προσοχή: Η C χειρίζεται τους χαρακτήρες σαν ακεραίους



Τύπος ακεραίου (int)

Αναπαράσταση (θετικών ή αρνητικών) ακεραίων αριθμών

- Παράδειγμα δήλωσης **int num;**
- Η περιοχή τιμών εξαρτάται από την αρχιτεκτονική του Η/Υ
- Συνήθως, αποθηκεύονται σε τόσα bits όσο το μήκος της **λέξης** του επεξεργαστή
 - Λέξη (word): αποτελεί την πληροφορία που μπορεί να διακινηθεί (I/O) σε ένα κύκλο μηχανής. Ταυτίζεται με το μέγεθος του διαύλου δεδομένων (data bus) του επεξεργαστή (8-bit, 16-bit, 32-bit, 64-bit).
- Λέξη 16-bit → περιοχή τιμών: -32767 ... +32768 (προσημασμένοι)
0 ... 65535 (χωρίς πρόσημο)

long int total; */* 32-bit αποθηκευτικός χώρος */*

unsigned int total; */* μεταβλητή total χωρίς πρόσημο */*



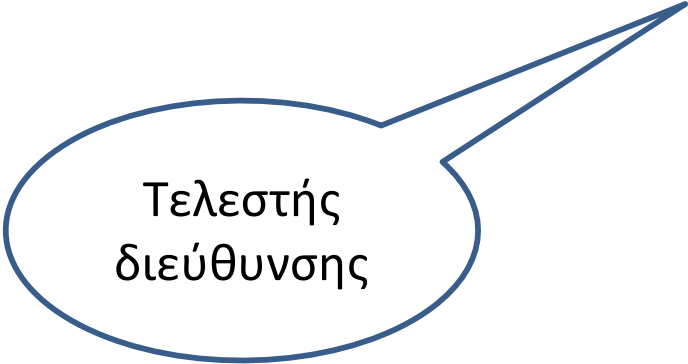
Τύπος ακεραίου (int)

Παράδειγμα εκτύπωσης

- Χρήση της **printf** μαζί με τους προσδιοριστές **%d**, **%x**, **%o** για εμφάνιση σε δεκαδική, δεκαεξαδική και οκταδική μορφή, αντίστοιχα
printf("dec=%d;, octal=%o;, hex=%x\n", num, num, num);

Παράδειγμα εισαγωγής

- Χρήση της **scanf** μαζί με τον προσδιοριστή **%d** και τον χαρακτήρα **&**
scanf("%d",&num);



Τελεστής
διεύθυνσης



Ο Απαριθμητικός τύπος (enum)

Συχνά, απαίτηση για νέους τύπους δεδομένων, οι οποίοι δεν περιλαμβάνονται στους ενσωματωμένους τύπους της γλώσσας

- Παράδειγμα: τρέχων μήνας (**current_month**)
- Πιθανές λύσεις: **int** ?, **char** ?

```
enum months {  
    JAN=1, FEB, MAR, APR, MAY, JUN  
    JUL, AUG, SEP, OCT, NOV, DEC  
};
```

```
enum months current_month;
```



Ο Απαριθμητικός τύπος (enum)

Συχνά, απαίτηση για νέους τύπους δεδομένων, οι οποίοι δεν περιλαμβάνονται στους ενσωματωμένους τύπους της γλώσσας

- Παράδειγμα: τρέχων μήνας (**current_month**)
- Πιθανές λύσεις: **int ?**, **char ?**

```
enum months {  
    JAN=1, FEB, MAR, APR, MAY, JUN  
    JUL, AUG, SEP, OCT, NOV, DEC  
};
```

```
enum months current_month;
```

Τύπος `boolean` (δεν είναι ενσωματωμένος στη C)

```
enum boolean {FALSE, TRUE};  
enum boolean success;
```


Τύποι πραγματικών (float, double)

Αναπαράσταση αριθμών με κλασματικό μέρος

- $1.258 * 10^6 = 1258000.0 = 1.258e+6 = 1.258E+6$
- $8.9 \times 10^{-4} = 0.00089 = 8.9e-4 = 8.9E-4$

Τύπος float → αριθμοί κινητής υποδιαστολής απλής ακρίβειας

- Έχει ακρίβεια τουλάχιστον 6-7 δεκαδικά ψηφία (αν και εξαρτάται από την υλοποίηση)

```
float val = 32000.0;
```



Τύποι πραγματικών (float, double)

Τύπος double → αριθμοί κινητής υποδιαστολής διπλής ακρίβειας

- Έχει διπλάσια ακρίβεια (14-15 δεκαδικά ψηφία)

```
double area;
```

Παράδειγμα εκτύπωσης

- Χρήση της **printf** μαζί με τους προσδιοριστές **%f**, **%e**, **%g** για εμφάνιση σε fixed point, εκθετική και επιλεγμένη από το σύστημα (συνήθως, αυτή με το μικρότερο μέγεθος) μορφή, αντίστοιχα

```
printf("%f μπορεί να γραφεί ως %e και ως %g\n", val, val,  
val);
```

- Εμφάνιση την οθόνη:

```
32000.000000 μπορεί να γραφεί ως 3.200000e+004 και ως  
32000
```

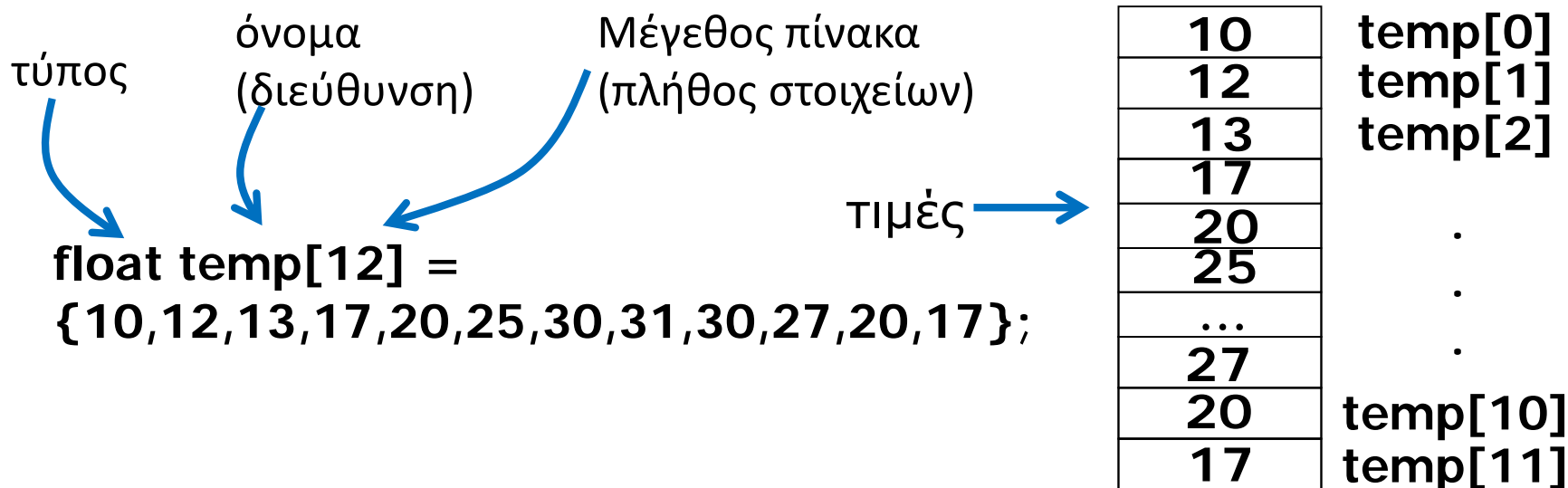


Τύπος πίνακα

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

Πετυχαίνεται γρήγορη πρόσβαση και προτιμούνται από άλλες δομές δεδομένων

Παράδειγμα: μέσες μηνιαίες θερμοκρασίες ενός έτους



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 */
```

- Παράλειψη ορισμού του μεγέθους του πίνακα

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

- Αρχικό (πρώτο) στοιχείο πίνακα

```
grade[0] → 6.5
```



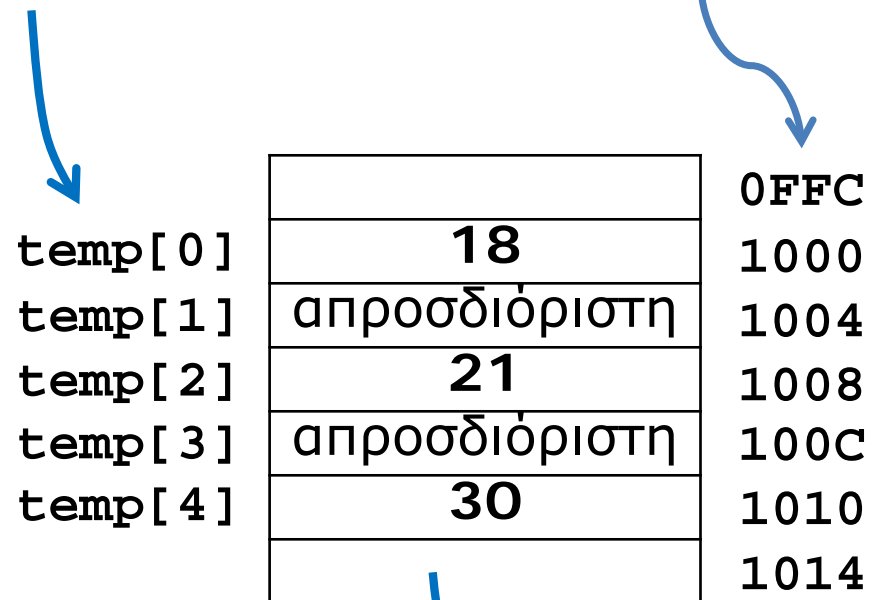
Αποθήκευση πινάκων στη μνήμη

Παράδειγμα

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

Στοιχεία
πίνακα

Διεύθυνση
μνήμης



← 4 bytes →



Πολυδιάστατοι πίνακες

Πίνακες των οποίων κάθε στοιχείο είναι πίνακας

- Παράδειγμα: πίνακας βαθμολογίας με 249 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[249][6];
```



Πολυδιάστατοι πίνακες

- Που αναφέρονται οι παρακάτω εκφράσεις;

`grade[0][5];`


`grade[1][0];`

`grade[199][3];`

Αποθήκευση

17	24	6
5	16	4
12	11	6

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

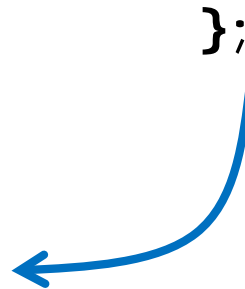


Πολυδιάστατοι πίνακες

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 4^ο

Απλά προγράμματα στη C

Απλά προγράμματα στη C

Συντακτικά και Σημασιολογικά Λάθη

Δομή προγράμματος

```
/* programma pou ypologizei athroisma
 * dyo akeraewn arithmwon */
#include <stdio.h>

void
main()
{ int numbera, numberb; /* inputs*/
  int sum;

  printf("Enter two numbers: ");
  scanf("%d%d",&numbera, &numberb);

  sum = numbera + numberb;

  printf("The sum of %d and %d is %d\n",
        numbera, numberb, sum);
}
```

→ Σχόλια

→ Βιβλιοθήκη για I/O

→ Συνάρτηση main

→ Δηλώσεις: ακέραιες μεταβλητές

→ Εντολές με συναρτήσεις stdio
→ Είσοδος δεδομένων (μεταβλητές έπονται του τελεστή διεύθυνσης &)

→ τελεστής πρόσθεσης (+)
→ δυαδικός τελεστής
→ τελεστής ανάθεσης (=)

→ Εντολή με συνάρτηση stdio
→ Εξαγωγή δεδομένων



Συντακτικά και Σημασιολογικά Λάθη

Κανόνες δημιουργίας ευανάγνωστου προγράμματος

- Αποφεύγετε (στις περισσότερες περιπτώσεις) ονόματα ενός χαρακτήρα, π.χ. `i`, `j`, `x`, `y`
 - Υπάρχουν όμως πολλές περιπτώσεις που χρησιμοποιούμε ονόματα τέτοιας μορφής
- Χρησιμοποιήστε εκφραστικά ονόματα
 - Αποφυγή παρερμηνειών, εύκολη αναγνώριση του αντικειμένου στο οποίο αναφέρεται το όνομα που δίνετε
 - Π.χ. δώστε ονόματα `length` και `width` για τις μεταβλητές «μήκος» και «πλάτος», αντίστοιχα. Στη συνέχεια, αν έχετε κάποια σχετική μεταβλητή, δώστε και σε αυτή ένα σχετικό όνομα (π.χ. `maximum_length` για «μέγιστο μήκος»)



Συντακτικά και Σημασιολογικά Λάθη

- Χρησιμοποιήστε μικρά γράμματα για ονόματα μεταβλητών
 - Χρησιμοποιούμε κεφαλαία γράμματα μόνο για μακροεντολές
- Δύο επιλογές για αντικείμενα που περιγράφονται από δύο ή περισσότερες λέξεις
 - Π.χ. `maximum_length` ή `maximumLength` για «μέγιστο μήκος»
 - Τηρήστε την παραδοχή σας σε όλο το πρόγραμμα



Παράδειγμα (καλής σύνταξης)

```
1  /* lab 01
2    Addition program */
3  #include <stdio.h>
4
5  int main(void)
6  {
7      int integer1, integer2, sum;          /* declaration */
8
9      printf( "Enter first integer\n" );    /* prompt */
10     scanf( "%d", &integer1 );           /* read an integer */
11     printf( "Enter second integer\n" );  /* prompt */
12     scanf( "%d", &integer2 );           /* read an integer */
13     sum = integer1 + integer2;          /* assignment of sum */
14     printf( "Sum is %d\n", sum );       /* print sum */
15
16     return 0; /* indicate that program ended successfully */
17 }
```



Παράδειγμα (κακής σύνταξης)

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      int x, y, s;
6
7      printf( "Enter first integer\n" );
8      scanf( "%d", &x );
9      printf( "Enter second integer\n" );
10     scanf( "%d", &y );
11     s = x + y;
12     printf( "Sum is %d\n", s );
13
14     return 0;
15 }
```

Σχόλια;

Ονόματα
μεταβλητών;



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

Copyright Πανεπιστήμιο Πατρών, Πολυτεχνική Σχολή, Τμήμα Μηχανολόγων & Αεροναυπηγών Μηχανικών, Νίκος Καρακαπιλίδης, Δημήτρης Σαραβάνος. Νίκος Καρακαπιλίδης, Δημήτρης Σαραβάνος. «Προγραμματισμός Η/Υ. Εισαγωγή στη C (Μέρος Πρώτο)». Έκδοση: 1.0. Πάτρα 2014. Διαθέσιμο από τη δικτυακή διεύθυνση: <https://eclass.upatras.gr/courses/MECH1207/>



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



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

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.



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

Οποιοδήποτε έργο στην παρούσα ενότητα, έχει δημιουργηθεί από τους διδάσκοντες του μαθήματος ή/και την Τμηματική Ομάδα Εργασίας και παρέχεται με την ίδια άδεια CC BY-NC-SA 4.0

