



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά
μαθήματα ΠΠ

Προγραμματισμός Η/Υ

Ενότητα 2β: Εισαγωγή στη C (Μέρος Δεύτερο)

Νίκος Καρακαπιλίδης, Καθηγητής

Δημήτρης Σαραβάνος, Καθηγητής

Πολυτεχνική Σχολή

Τμήμα Μηχανολόγων & Αεροναυπηγών Μηχανικών

Σκοποί ενότητας

- Κατανόηση της έννοιας του Τελεστή
- Κατανόηση της χρήσης των παρενθέσεων στο πρόγραμμα



Περιεχόμενα ενότητας

- Η έννοια του Τελεστή
- Εκφράσεις
- Μετατροπές Τύπων
- Οι τελεστές της C
- Είσοδος / Έξοδος



Μέρος 1^ο

Η έννοια του Τελεστή

Η έννοια του Τελεστή

Ορισμός του Τελεστή

Η έννοια του τελεστή

Τελεστής (operator) είναι ένα σύμβολο ή μια λέξη μιας γλώσσας προγραμματισμού που αναπαριστά κάποια συγκεκριμένη διαδικασία

- Η διαδικασία αυτή εκτελείται πάνω σε ένα ή περισσότερα δεδομένα
- Τελεστέοι (operands)
- Παράδειγμα: `num_1 > num_2`



Κατηγορίες τελεστών

- Αριθμητικοί $+ - * / \%$
- Λογικοί $\&\& \|\| !$
- Συσχετιστικοί $> >= == < <= !=$
- Διαχείρισης των bits ενός byte $>> \& | ^ \sim$
- Διαχείρισης μνήμης $\& [] . ->$



Μέρος 2^ο

Εκφράσεις

Εκφράσεις

Προτεραιότητα
Προσεταιριστικότητα
Χρήση παρενθέσεων

Εκφράσεις

Γενική περίπτωση

- Αποτελείται από έναν ή περισσότερους τελεστέους και προαιρετικά από έναν ή περισσότερους τελεστές
- Τελεστέοι \rightarrow μεταβλητές, σταθερές ή κλήσεις συναρτήσεων
- **Κάθε έκφραση έχει μια τιμή**



Εκφράσεις

Παραδείγματα απλών εκφράσεων

- **8**
- **count**
- **func(x)**
- **MAX_NUM**

Παραδείγματα σύνθετων εκφράσεων

- **8 * 24**
- **count + 4**
- **func(x) / 3**
- **(count + 32) / MAX_NUM**



Προσεταιριστικότητα (associativity)

Προσεταιριστικότητα (associativity)

- Καθορισμός σειράς εφαρμογής περισσότερων τελεστών στο ίδιο επίπεδο (πολλά στιγμιότυπα του τελεστή)
 - $+$, $-$, $*$ και $/$ είναι αριστερής προσεταιριστικότητας (εφαρμογή από αριστερά προς τα δεξιά)
 - Τελεστής ανάθεσης είναι δεξιάς προσεταιριστικότητας

num_1 = num_2 = 5

Χρήση παρενθέσεων για προσδιορισμό συγκεκριμένης σειράς εφαρμογής τελεστών ή/και αύξηση αναγνωσιμότητας μιας έκφρασης

- Πρώτα υπολογίζονται οι εσωτερικές παρενθέσεις

Προσοχή: Χρησιμοποιήστε παρενθέσεις όπου υπάρχει αβεβαιότητα



Προσεταιριστικότητα (associativity)

Παραδείγματα

- $2 - 3 * 4$
- $(2 - 3) * 4$
- $2 - (3 * 4)$

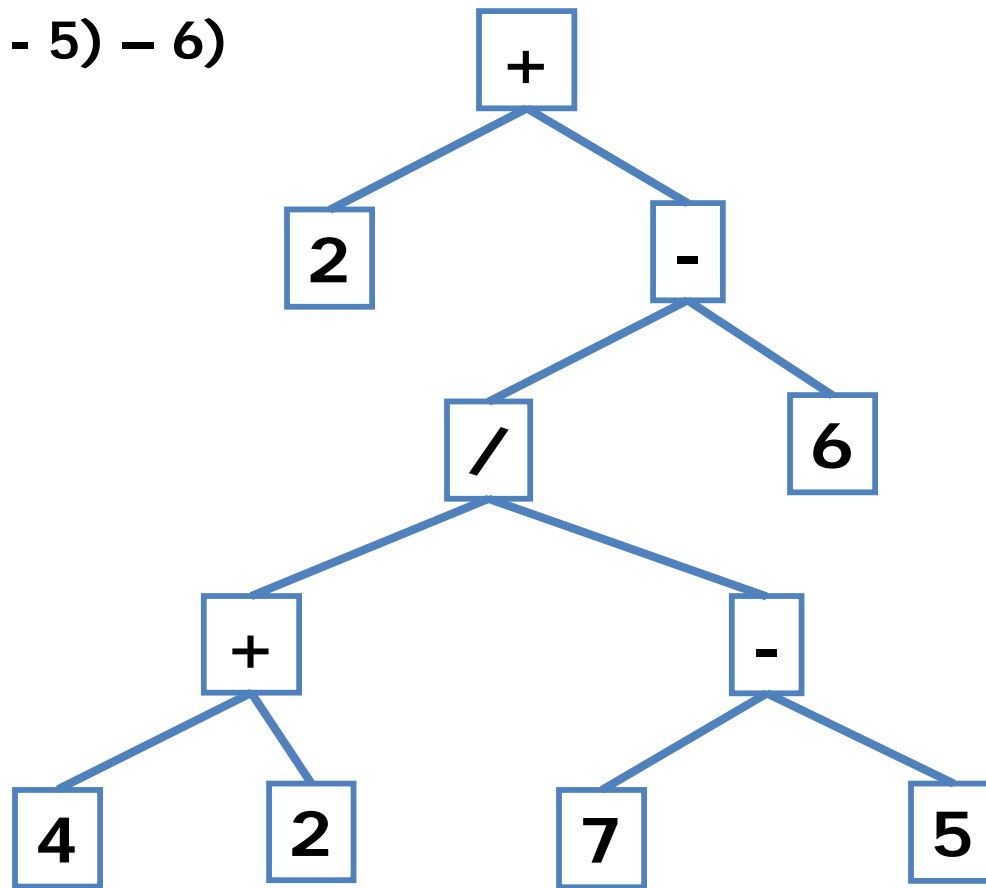
- $x + y + z * w$
- $(x + (y + (z * w)))$

- $2 + ((4 + 2) / (7 - 5) - 6)$



Δέντρο αφηρημένης σύνταξης

$$2 + ((4 + 2) / (7 - 5) - 6)$$



Μέρος 3^ο

Μετατροπές Τύπων

Μετατροπές Τύπων

Υπονοούμενες και ρητές μετατροπές

Μετατροπές τύπων

Η C επιτρέπει (με κάποιους περιορισμούς) την ανάμειξη αριθμητικών τύπων σε εκφράσεις

- Ο μεταγλωττιστής μετατρέπει αυτόματα τους τελεστέους έτσι ώστε να είναι του ιδίου τύπου

Υπονοούμενες μετατροπές (implicit conversions)

- Εκτελούνται αυτόματα (ο «στενότερος» τύπος μετατρέπεται στον «ευρύτερο» χωρίς απώλεια πληροφορίας)
- `char < int < long < float < double`



Μετατροπές τύπων

- Υπολογισμός αριθμητικών εκφράσεων (για όλους τους compilers της C)
 - `char` → `int`
 - `float` → `double`

Ρητές μετατροπές (explicit conversions)

- Casting: `j = (float) 2`



Μετατροπές τύπων

Παραδείγματα

- Τι τιμή δίνει η παρακάτω έκφραση;

3.0 + 1/2

- Τι τιμές παίρνουν οι f και i στο παρακάτω παράδειγμα;

int i;

float f;

f = i = 3.3



Πρόταση

Μια πλήρης εντολή (command)

- Κάθε πρόταση έχει στο τέλος της το ελληνικό ερωτηματικό (;)

Κατηγορίες προτάσεων

- Απλές vs. σύνθετες (χρήση { })

- Προτάσεις δηλώσεων

```
int number;
```

- Προτάσεις κλήσης συναρτήσεων

```
printf("Hello World");
```

- Προτάσεις ροής ελέγχου

```
if (a > b) then a else b;
```

- Προτάσεις αναθέσεων

```
number = 35;
```

- Προτάσεις προεπεξεργαστή (χωρίς ;)

```
#include <stdio.h>
```

```
#define PI
```

```
3.141592654
```

- Μηδενική πρόταση

```
;
```



Μέρος 4^ο

Οι τελεστές της C

Οι τελεστές της C

Αριθμητικοί, ανάθεσης, συσχετιστικοί,
λογικοί, μετατροπής τύπου, sizeof,
υποθετικός

Οι τελεστές της C

Αριθμητικοί τελεστές

- Δυαδικοί $+$, $-$, $*$, $/$, $\%$ (τελεστής υπολοίπου)

x	y	x/y	x%y
4	2	2	0
4	3	1	1

- Μοναδιαίος $-$



Οι τελεστές της C

- Μοναδιαίοι αύξησης και μείωσης $++$, $--$
 - Προπορευόμενοι, π.χ. $++X$

→ η αύξηση ή η ελάττωση γίνεται πριν να χρησιμοποιηθεί η τιμή της μεταβλητής στον υπολογισμό της τιμής της έκφρασης στην οποία συμμετέχει
 - Παρελκόμενοι, π.χ. $X++$

→ η αύξηση ή η ελάττωση γίνεται μετά τον υπολογισμό της τιμής της έκφρασης στην οποία συμμετέχει



Οι τελεστές της C

Πίνακας 1: Αριθμητικοί τελεστές αύξησης και μείωσης

Πρόταση	Τιμή x	Τιμή y
<code>int x=10, y=20;</code>	10	20
<code>++x;</code>	11	20
<code>y = --x;</code>	10	10
<code>y = x-- + y;</code>	9	20
<code>y = y - x++;</code>	10	11



Οι τελεστές της C

Τελεστές ανάθεσης

- Απλός
 - `count = count + 1;`
 - `new_number = old_number;`
- Σύνθετοι
 - `x *= 10;` \rightarrow `x = x * 10;`
 - `x *= y + 1;` \rightarrow `x = x * (y + 1);`



Οι τελεστές της C

Συσχετιστικοί τελεστές (ή σύγκρισης)

$<$, $>$, $<=$, $>=$, $==$ (ελέγχου ισότητας), $!=$ (διάφορο)

- Το αποτέλεσμα έκφρασης που περιέχει τέτοιο τελεστή είναι αληθές (true) ή ψευδές (false)
- Στη C, η τιμή είναι ακέραια 0 για false και 1 για true

Λογικοί τελεστές

- Χρήση: συνένωση δύο ή περισσότερων εκφράσεων σύγκρισης (δημιουργία λογικών εκφράσεων)
- $\&\&$ (AND), $||$ (OR), $!$ (NOT)

$(x < y) \&\& (y < z)$



Οι τελεστές της C

Τελεστής sizeof

- Επιστρέφει τον αριθμό των bytes που η τιμή μιας έκφρασης ή ένας τύπος δεδομένων καταλαμβάνει στη μνήμη

int age[5]; → **sizeof(age)** δίνει τιμή **20** (γιατί;)

→ **sizeof(age[0])** δίνει τιμή **4** (γιατί;)



Οι τελεστές της C

Υποθετικός τελεστής

- Γενική μορφή: έκφραση_1 ? έκφραση_2 : έκφραση_3
- Παράδειγμα: η έκφραση

x > z ? x : z;

έχει τιμή **x** αν το **x > z** είναι αληθές, αλλιώς έχει τιμή **z**



Οι τελεστές της C

Τελεστής μετατροπής τύπου (ή cast)

- Ρητή μετατροπή τύπων
- Παράδειγμα:

int num; → (double)num



Μέρος 5^ο

Είσοδος / Έξοδος

Είσοδος / Έξοδος

Μορφοποίηση

Είσοδος / Έξοδος

Συναρτήσεις εισόδου / εξόδου (input / output, I/O)

- Μια συνάρτηση εισόδου «αντιγράφει» δεδομένα από μονάδα εισόδου σε χώρο στην μνήμη
 - **scanf, getchar**
- Μια συνάρτηση εξόδου εκτυπώνει πληροφορίες που είναι αποθηκευμένες στην μνήμη σε μονάδα εξόδου
 - **printf, putchar**
- Η βιβλιοθήκη **stdio** περιέχει βασικές συναρτήσεις εισόδου και εξόδου



printf

Σύνταξη

printf(μορφή εξόδου);

printf(μορφή εξόδου, λίστα εκφράσεων);



printf

παράμετροι (arguments)

```
int count = 37;  
printf("The total count is %d\n", count);
```

Σημασία:

αντικατέστησε το όρισμα με την τιμή της έκφρασης από τη λίστα εκφράσεων

string
(τυπώνεται
ως έχει)

όρισμα

ειδικός
χαρακτήρας

μεταβλητή

%d → ακέραια τιμή, %c → χαρακτήρας,
%f → κινητής υποδιαστολής μονής ακρίβειας,
%e → κινητής υποδιαστολής διπλής ακρίβειας

Προσοχή: Αριθμός ορισμάτων πρέπει να είναι ίδιος με το μέγεθος της λίστας εκφράσεων. Επίσης, η σειρά και οι τύποι πρέπει να ταιριάζουν

Ειδικοί Χαρακτήρες

Σημασία

`\n` μετέφερε δρομέα στην επόμενη γραμμή

`\t` μετέφερε δρομέα στην επόμενη δεξιά στήλη

`\"` εκτύπωσε "

`\\` εκτύπωσε \



Μορφοποίηση (formatting)

Σύνταξη

`%nd`

Σημασία

- χρησιμοποίηση **n** θέσεων για την εκτύπωση του ακέραιου αριθμού. Εάν ο αριθμός έχει λιγότερα από **n** ψηφία, τοποθετούνται κενά στα αριστερά (δεξιά ευθυγράμμιση). Εάν ο αριθμός έχει περισσότερα ψηφία, εκτυπώνονται όλα



Μορφοποίηση (formatting)

```
printf("\n%3d%3d\n\n%2d %3d", 1, 22, 33, -444);
```

		1		2	2		
3	3		-	4	4	4	



Μορφοποίηση (formatting)

Σύνταξη

%n.mf

Σημασία: χρησιμοποίηση **n** θέσεων για την εκτύπωση του πραγματικού αριθμού (υπολογίζεται και η υποδιαστολή) και **m** θέσεων για τα δεκαδικά ψηφία

Σύνταξη

%.mf

```
printf("%4.2f %6.3f %6.2f", 4.4, 22.1, 466.);
```

```
4.40 22.100 466.00
```



scanf

Σύνταξη:

scanf(μορφή εισόδου, λίστα διευθύνσεων μεταβλητών);

```
int number_1, number_2;  
scanf("%d%d", &number_1, &number_2);
```

ορίσματα

**διευθύνσεις
μεταβλητών**

Σημασία: Δηλώνουμε τον τύπο και τη σειρά των τιμών που θα εισαχθούν (ορίσματα ανάλογα με την printf)



Τελεστής διεύθυνσης και scanf

Σύνταξη: **&** <όνομα μεταβλητής>

Σημασία: δίνει τη διεύθυνση της μεταβλητής

&number_1

Η scanf αποθηκεύει τιμές στις μεταβλητές χρησιμοποιώντας τις διευθύνσεις τους



Τελεστής διεύθυνσης και scanf

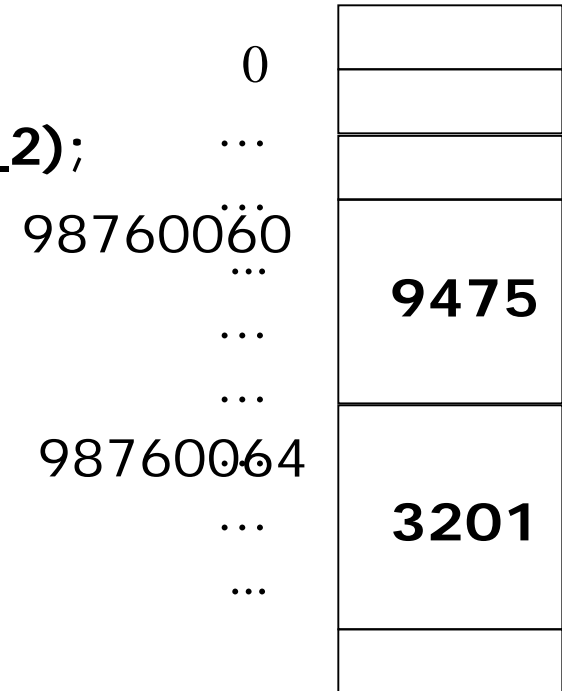
```
int number_1, number_2;  
  
printf("Enter two integer numbers: ");  
scanf("%d%d",&number_1, &number_2);  
  
printf("The two numbers entered are %d %d\n\n",  
      number_1, number_2);
```



Τελεστής διεύθυνσης και scanf

```
int number_1, number_2;
```

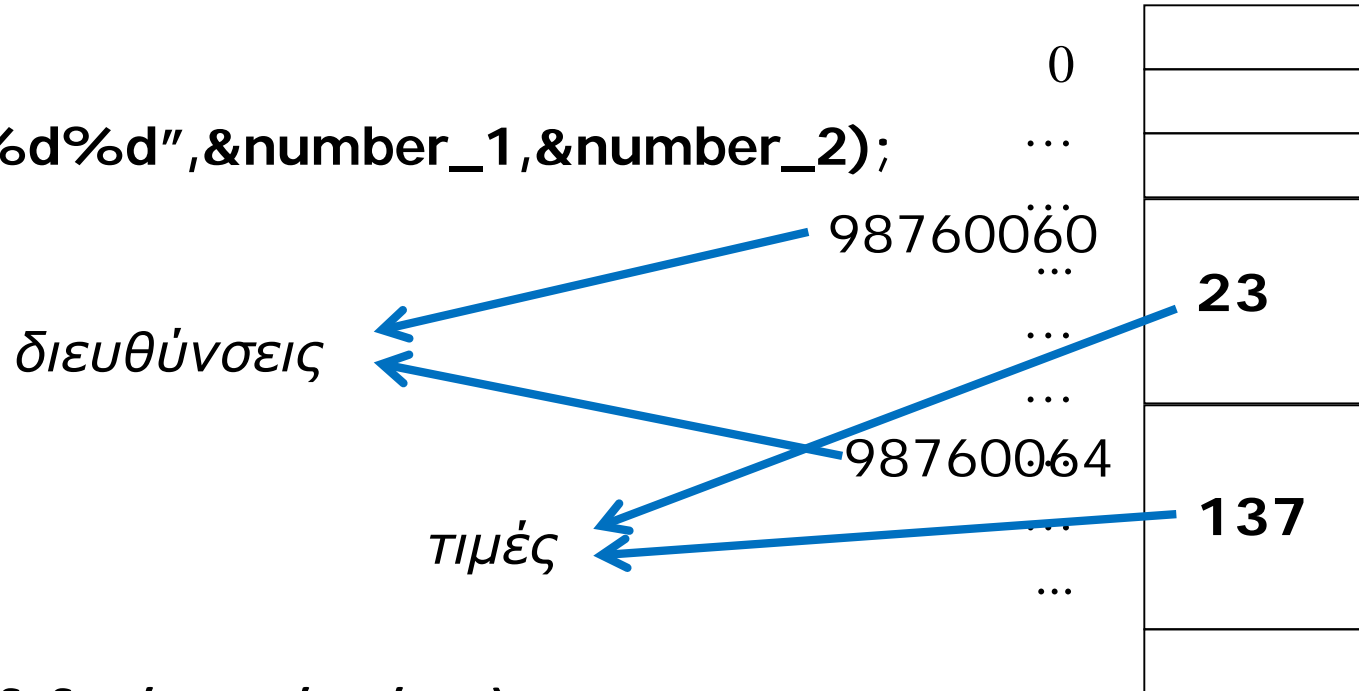
```
scanf("%d%d",&number_1,&number_2);
```



Τελεστής διεύθυνσης και scanf

```
int number_1, number_2;
```

```
scanf("%d%d",&number_1,&number_2);
```



→ 23 137 (δεδομένα από χρήστη)



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 Δεύτερο)». 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

