Διάλεξη 6 - Εντολές και Ροή Ελέγχου

Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

Εισαγωγή στον Προγραμματισμό

Θανάσης Αυγερινός

Ανακοινώσεις / Διευκρινίσεις

- Αναπλήρωση Διάλεξης την Πέμπτη, 31 Οκτωβρίου 7-9μμ
- Δεν καταλαβαίνω κάτι που φαίνεται σημαντικό; Ρωτάω τώρα / στο διάλειμμα!
- Πως μπορώ να τεστάρω το trolley;

Οι (περσινοί) πρωτοετείς μας προκρίθηκαν στο GRCPC!

RANK	TEAM	SCORE	A	В	C	D	E	F	G	Н
1	The ICPC Greece Regional Competition Should I Stay or Should I Code National Technical University of Athens	8 1196	113 1 try	64 1 try	211 4 tries	174 3 tries	127 1 try	70 1 try	160 2 tries	137 2 tries
2	GeeksforGreeks National Technical University of Athens									
3	CyPros National Technical University of Athens	7 909								
4	HUAdraticComplexity Harokoplo University of Athens									
5	SleepIsOverrated University of Piraeus									
6	AC/DC National Technical University of Athens	4 361		32 3 tries	4 tries	5 tries	70 1 try	199 1 try	20 1 try	
7	0x8BADF00D National and Kapodistrian University of Athens	4 393		99 1 try	187 1 try	5 tries		81 1 try	26 1 try	
8	GetElementPtr National Technical University of Athens	4 751								
9	Acoustics National Technical University of Athens									
10	Killswitch Aristotle University of Thessaloniki									



Τεστάροντας το trolley

```
$ time ./trolley < text/trolley.txt > output
real 0m0.082s
user 0m0.005s
sys 0m0.078s
$ grep -c "Go right" output
4995
$ grep -c "Go left" output
5005
```

Testcase διαθέσιμο στο https://github.com/progintro/data/blob/main/text/trolley.txt

Την Προηγούμενη Φορά

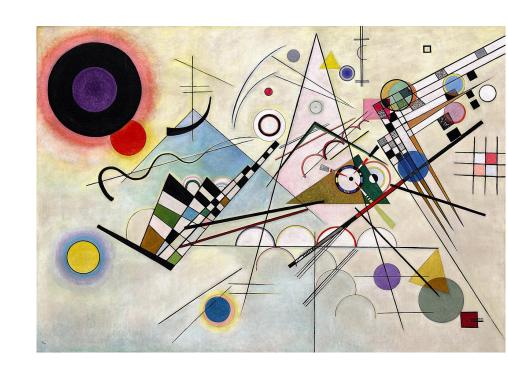
- Τελεστές
- Εκφράσεις και Εντολές



Σήμερα

- Εντολές συνέχεια
- Ροή Ελέγχου
- Παραδείγματα Εντολών &

Προβλημάτων



Κατηγοριοποίηση Εντολών

- 1. Κενές Εντολές
- 2. Εντολές Έκφρασης
- 3. Σύνθετες Εντολές
- 4. Εντολές Συνθήκης
- 5. Εντολές Επανάληψης

Κενή Εντολή (Empty / Null Statement)

Η κενή εντολή δεν εκτελεί τίποτε (no-operation ή no-op) και έχει την μορφή:

```
; // null statement
;;;;; // 5 null statements
```

Η χρησιμότητά της φαίνεται σε συνδυασμό με άλλες εντολές.

Σημείωση: χρησιμοποιούμε semicolon (;) ως διαχωριστικό εντολών - για να δείξουμε που τελειώνει η εντολή μας.

Εντολή Έκφρασης (Expression Statement)

Εντολή έκφρασης στην C ονομάζουμε μια έκφραση που τελειώνει με semicolon (;). Παραδείγματα expression statements:

```
x = 4;

y = 7;

z = ++y;

y = z - (x++);

z = x - (--y);
```

Ποια η τιμή των x, y, z μετά την εκτέλεση αυτών των εντολών έκφρασης;

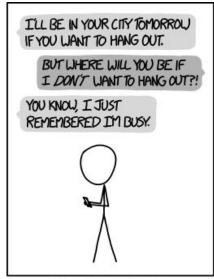
Σύνθετη Εντολή (Compound Statement / Block)

Σύνθεση εντολή ή αλλιώς block, λέγεται μια σειρά από εντολές όταν περικλείονται από αγκύλες { }. Παράδειγμα:

```
{
x = 4;
y = 7;
z = ++y;
y = z - (x++);
z = x - (--y);
```

Εντολές Ροής Ελέγχου (Control Flow Statements)

Η ροή ελέγχου, δηλαδή η σειρά με την οποία θα εκτελεστούν οι εντολές σε ένα πρόγραμμα καθορίζεται από τις εντολές ροής ελέγχου. Συνήθως οπτικοποιείται με ένα διάγραμμα.



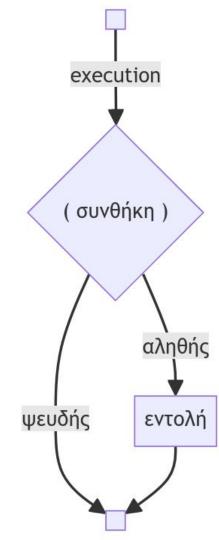
WHY I TRY NOT TO BE PEDANTIC ABOUT CONDITIONALS.

Εντολή if (if Statement)

Η εντολή if εκτελεί μια εντολή αν μια λογική συνθήκη είναι αληθής. Γενική μορφή:

```
if ( συνθήκη )
εντολή
```

Παραδείγματα:



Εντολή if (if Statement)

Η εντολή if εκτελεί μια εντολή αν μια λογική συνθήκη είναι αληθής. Γενική μορφή:



Παραδείγματα:

year++;

if (year > 1) { year++;

(συνθήκη) Οι παρενθέσεις είναι υποχρεωτικές! αληθής ψευδής εντολή

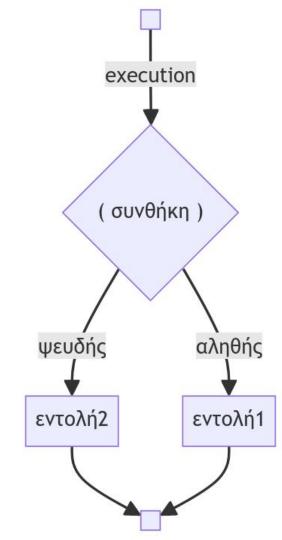
execution

Εντολή if-else (if-else Statement)

Επέκταση της εντολής if προκειμένου να εκτελέσουμε μια άλλη εντολή αν η λογική συνθήκη είναι ψευδής.

```
if ( συνθήκη )
εντολή1
else
```

εντολή2



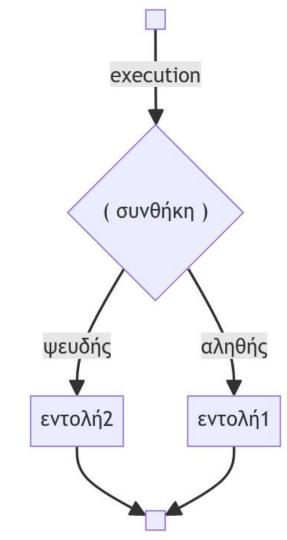
Εντολή if-else (if-else Statement)

Παραδείγμα:

Θα μπορούσαμε να "ζήσουμε" χωρίς else;

Υπάρχει εναλλακτικός τρόπος να γράψουμε το παραπάνω;

Τι κάνουμε όταν έχουμε πάνω από δύο συνθήκες;



Εμφωλευμένες Εντολές if (Nested if)

Πολλές φορές χρειάζεται να ελέγξουμε πάνω από μια συνθήκες και χρησιμοποιούμε εμφωλευμένες if εντολές:

```
lab = -1;
if (year < 1)
     { /* empty block */ }
else if (year == 1)
     lab = 20;
else
     lab = 0;</pre>
```

To <u>Dangling else</u> πρόβλημα

Πολλές εμφωλευμένες εντολές if μπορεί να οδηγήσουν σε προβλήματα αμφισημίας (κάτι που δεν μας αρέσει καθόλου στο computer science). Είναι τα δύο προγράμματα ισοδύναμα;

Το πρόβλημα με το <u>Dangling else</u>

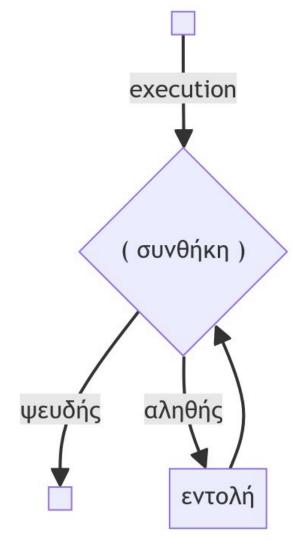
Πολλές εμφωλευμένες εντολές if μπορεί να οδηγήσουν σε προβλήματα αμφισημίας (κάτι που δεν μας αρέσει καθόλου στο computer science). Είναι τα δύο προγράμματα ισοδύναμα;

Αν δεν είμαστε σίγουροι χρησιμοποιούμε αγκύλες { } για να υποδείξουμε τα όρια των σύνθετων εντολών

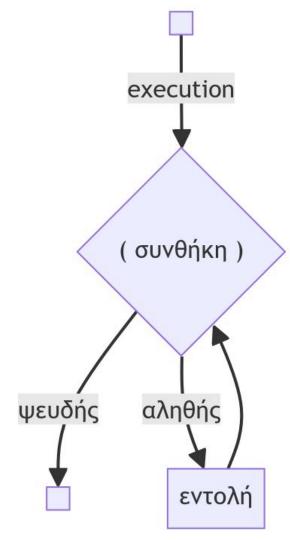
Δομές Επανάληψεις / Loops / Βρόχοι

Η εντολή while επαναλαμβάνει μια άλλη εντολή όσο η λογική συνθήκη είναι αληθής.

while (συνθήκη) εντολή



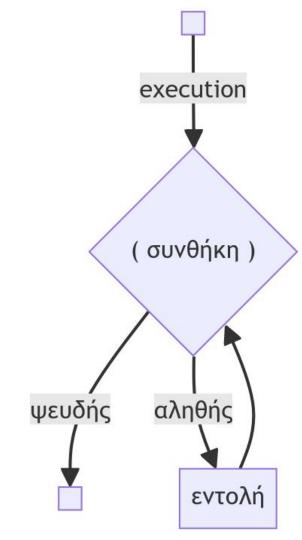
```
Παράδειγμα #1: Τι κάνει το παρακάτω πρόγραμμα;
           int i = 0;
           int sum = 0;
           while (i < 42) {</pre>
              sum += i;
              i++;
           printf("%d %d\n", i, sum);
```



Παράδειγμα #2: Τι κάνει το παρακάτω πρόγραμμα;

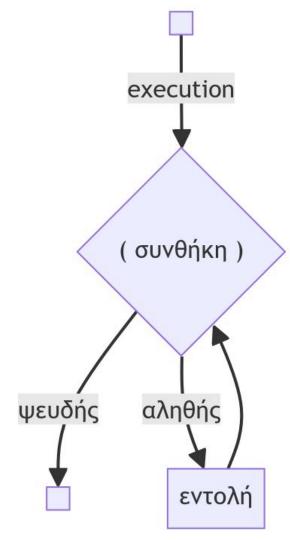
while (1);

Υπάρχει κάποια χρησιμότητα στο να έχουμε μια λογική συνθήκη που είναι πάντα αληθής;



Παράδειγμα #3: Τι κάνει το παρακάτω πρόγραμμα;

```
i = 0;
while (i < 42);
printf("%d\n", i++);</pre>
```



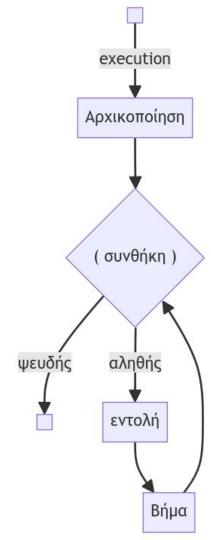
Εντολή for (for Statement)

Η εντολή for: αρχικοποιεί μεταβλητές, επαναλαμβάνει μια εντολή όσο η λογική συνθήκη είναι αληθής και στο τέλος κάθε επανάληψης εκτελεί το βήμα. Γενική μορφή:

```
for ( αρχικοποίηση ; συνθήκη ; βήμα )
εντολή
```

Παράδειγμα:

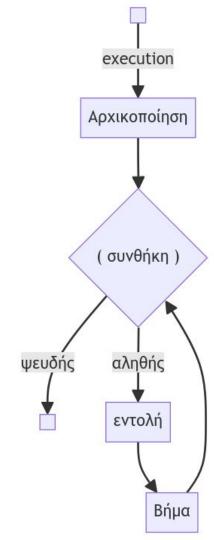
```
int i;
for ( i = 0 ; i < 100 ; i++ )
    printf("Hello world\n");</pre>
```



Εντολή for (for Statement)

Τι κάνει το παρακάτω πρόγραμμα;

for (;;)

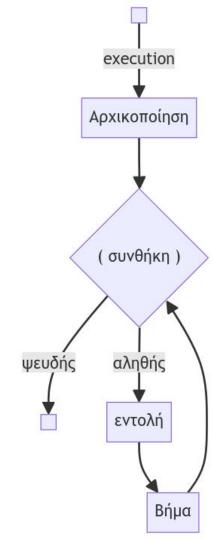


Εντολή for (for Statement)

Γενικό παράδειγμα:

```
int i;
for ( i = 0 ; i < N ; i++ ) {
    printf("%d\n", i);
}</pre>
```

Πόσες φορές θα εκτελεστεί το block εντολών μέσα στην for; Θα τυπωθεί το N; Τι θα συμβεί αν αλλάξω την αρχικοποίηση ή το βήμα;

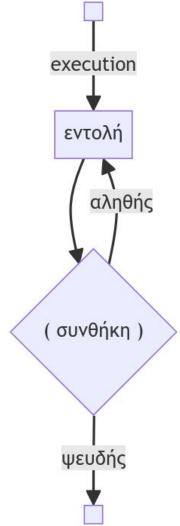


Η εντολή do-while τρέχει πρώτα μια φορά την εντολή και στην συνέχεια ελέγχει την λογική συνθήκη για το αν χρειάζεται άλλη επανάληψη.

do

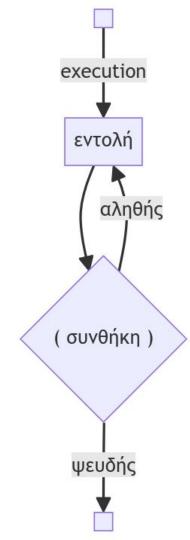
εντολή

while (συνθήκη);



Παράδειγμα: Τι κάνει το παρακάτω πρόγραμμα;

```
i = 0;
do
   i++:
while (i < 42);
printf("%d\n", i);
```





Για την Επόμενη Φορά

- Από τις σημειώσεις του κ. Σταματόπουλου συνιστώ να έχετε καλύψει τα πάντα μέχρι την σελίδα 62.
- Control flow
- <u>Conditional statements</u> in programming languages
- While loops
- For loops

Ευχαριστώ και καλό Σαββατοκύριακο εύχομαι!

Keep (or Start!) coding;)