Εισαγωγή στην Πληροφορική & στον Προγραμματισμό

Αρχές Προγραμματισμού Η/Υ (με τη γλώσσα C)

Διάλεξη #2

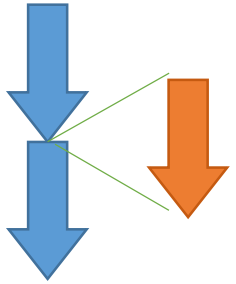
Παρασκευή, 12 Μαρτίου 2021

Παναγιώτης Παύλου

c-programming@mail.ntua.gr

Αρχικά είδαμε...

- Ροή εκτέλεσης προγράμματος



- Κλήση συναρτήσεων

```
printf("Hello!\n");
```

ή

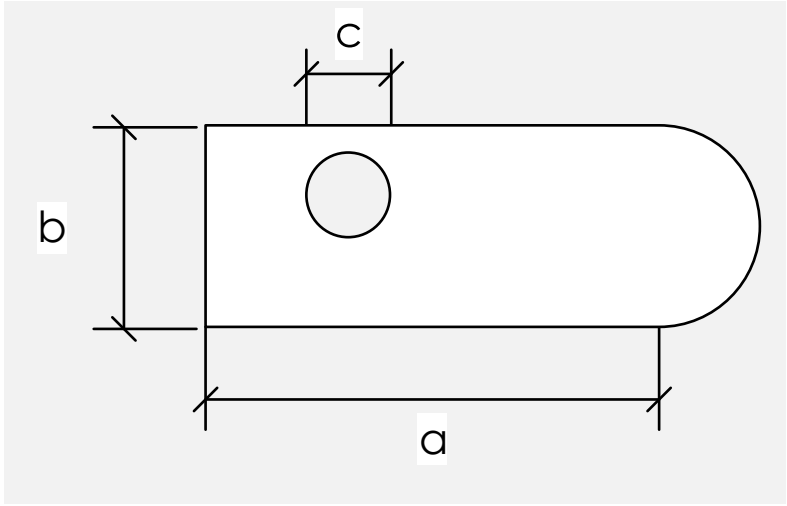
```
x = fabs(y);
```

- Βιβλιοθήκες
  - math
  - stdlib

Εφαρμογή #1

Να γραφτεί ένα πρόγραμμα το οποίο να:

1. υπολογίζει το εμβαδό του σχήματος
2. ύστερα να το στρογγυλοποιεί σε στο αμέσως μεγαλύτερο πολλαπλάσιο του 100 προς τα πάνω
3. κατόπιν να υπολογίζει υπεραπλουστευτικά πόσα κουτάκια μπογιάς που το κάθε ένα καλύπτει συγκεκριμένη επιφάνεια θα χρειάζονταν για να καλυφθεί η επιφάνεια του σχήματος
4. τέλος να υπολογίζει πόσο μέρος του «τελευταίου κουτιού» περισσεύει και να το παρουσιάζει ως αριθμό και ως ποσοστό του περιεχομένου του.




```
#include <stdio.h>
#include <math.h>
int main() {
    double a = 104, b = 401, c = exp(1);
    double Ebox = a * b;
    double Er = M_PI * pow( b/2.0, 2.0 ) / 2.0;
    double Ecrc = M_PI * pow( c/2.0, 2.0 );
    double A = Ebox + Er - Ecrc;
    double factor = pow( 10, 2 );
    double Ar = ceil( A / factor ) * factor;
    printf("%lf -> %lf\n", A, Ar);
    double stickerSize = 123;
    long stickers = (long)floor(A/stickerSize)+1;
    printf("Stickers: %ld\n", stickers);
    double remainder = stickers * stickerSize - A;
    printf("Loss: %lf (%.2lf%%)\n",
           remainder,
           100 * remainder / stickerSize);
    return 0;
}
```

Ερωτήσεις?

- Διαβάστε τις σημειώσεις, διαβάστε τις διαφάνειες και δείτε τα videos **πριν** ρωτήσετε
- **Συμβουλευτείτε** τη σελίδα ερωταποκρίσεων του μαθήματος

  https://qna.c-programming.allos.gr
- **Στείλτε** τις ερωτήσεις σας πριν και μετά το μάθημα στο

  c-programming@allos.gr

- Εάν έχετε **πρόβλημα** με κάποιο κώδικα στείλτε μαζί τον κώδικα και τα μηνύματα λάθους από το CLion ως κείμενα με copy/paste. Εάν θεωρείτε ότι επιπλέον βοηθά και ένα στιγμιότυπο οθόνης, είναι καλοδεχούμενο.
- Επαναλαμβάνουμε : Μην στείλετε ποτέ κώδικα ως εικόνα μας είναι παντελώς άχρηστος!

και ακολούθως …

- Ορισμός συναρτήσεων

```
double squareOf(double x) {
  return x*x;
}
```

- Δήλωση συναρτήσεων

```
double squareOf(double x);
```

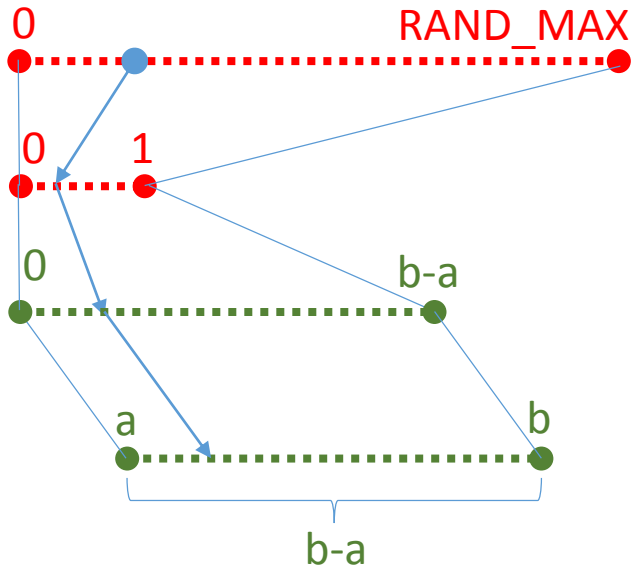
Τυχαίος αριθμός από Α έως Β

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

double random(double from, double to);

int main() {
    printf("%lf\n", random(3.5, 8.8));
    return 0;
}

double random(double a, double b) {
    double R = rand();
    R = R / RAND_MAX;
    double helper = r * (b - a);
    return a + helper;
}
```



Να δημιουργηθεί μία συνάρτηση η οποία να:

επιστρέφει έναν τυχαίο πραγματικό αριθμό από α ως β.

Ερωτήσεις?

- Διαβάστε τις σημειώσεις, διαβάστε τις διαφάνειες και δείτε τα videos **πριν** ρωτήσετε
- **Συμβουλευτείτε** τη σελίδα ερωταποκρίσεων του μαθήματος

  https://qna.c-programming.allos.gr
- **Στείλτε** τις ερωτήσεις σας πριν και μετά το μάθημα στο

  c-programming@allos.gr

- Εάν έχετε **πρόβλημα** με κάποιο κώδικα στείλτε μαζί τον κώδικα και τα μηνύματα λάθους από το CLion ως κείμενα με copy/paste. Εάν θεωρείτε ότι επιπλέον βοηθά και ένα στιγμιότυπο οθόνης, είναι καλοδεχούμενο.
- Επαναλαμβάνουμε : Μην στείλετε ποτέ κώδικα ως εικόνα μας είναι παντελώς άχρηστος!

τέλος ...

- Προεπεξεργαστής

```
#define THIS_YEAR 2021
#define AplusB (a+b)
```

- και

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

ή

```
#include "my-header.h"
```

Που περιέχει “defines” και δηλώσεις συναρτήσεων.

Παράδειγμα

```
#include <stdio.h>

#define BEGIN {
#define END }

#define ONEpTHREE 1+3

#define PROD(x,y) ((x)*(y))

int main()
BEGIN
    printf("Result: %d\n", PROD(ONEpTHREE, 2+4));
    // Προσοχή! Η παραπάνω παράσταση γίνεται ((1+3)*(2+4)) πριν το build
    return 0;
END
```

Ερωτήσεις?

- Διαβάστε τις σημειώσεις, διαβάστε τις διαφάνειες και δείτε τα videos **πριν** ρωτήσετε
- **Συμβουλευτείτε** τη σελίδα ερωταποκρίσεων του μαθήματος

  https://qna.c-programming.allos.gr
- **Στείλτε** τις ερωτήσεις σας πριν και μετά το μάθημα στο

  c-programming@allos.gr

- Εάν έχετε **πρόβλημα** με κάποιο κώδικα στείλτε μαζί τον κώδικα και τα μηνύματα λάθους από το CLion ως κείμενα με copy/paste. Εάν θεωρείτε ότι επιπλέον βοηθά και ένα στιγμιότυπο οθόνης, είναι καλοδεχούμενο.
- Επαναλαμβάνουμε : Μην στείλετε ποτέ κώδικα ως εικόνα μας είναι παντελώς άχρηστος!

Σχόλια + Μορφοποίηση = Λιγότερα λάθη

Το ζήτημα της μείωσης των σφαλμάτων εξαρχής επιτρέπει τη μείωση του χρόνου του debugging. Σε αυτό συμβάλλουν πολύ:

Τα **σχόλια** στον κώδικα, δηλαδή κείμενα που δεν εκτελούνται ή χρησιμοποιούνται, αλλά αποτελούν «σημειώσεις» του προγραμματιστή για να μπορεί να επανέλθει ο ίδιος στον κώδικά του μετά από καιρό ή να συνεργαστεί με άλλον χωρίς να εξηγεί τα πάντα.

Τα ελάχιστα σχόλια είναι κυρίως οδηγίες χρήσης για τις συναρτήσεις, αλλά και επεξηγήσεις του ρόλου της κάθε μεταβλητής.

Η σωστή **μορφοποίηση** (είδαμε πως μας βοηθά το IDE) επιτρέπει την εύκολη αναγνώριση των block εντολών, κάνει ευανάγνωστες τις παραστάσεις και γενικά βοηθά στην γρηγορότερη κατανόηση του κώδικα που εμφανίζεται στην οθόνη.

Σχόλια – Comments

Τα σχόλια είναι δύο τύπων:

- Τα σχόλια **γραμμής** ή line comments
  που ξεκινούν με // και εκτείνονται μέχρι το τέλος της γραμμής. Συνήθως χρησιμοποιούνται για να σχολιάσουν ένα συγκεκριμένο σημείο του προγράμματος
- Τα σχόλια **περιοχής** ή block comments
  που ξεκινούν με /* και εκτείνονται μέχρι το */ Συνήθως χρησιμοποιούνται για να σχολιάσουν μία συνάρτηση ή για να «απενεργοποιήσουν» μία περιοχή του κώδικα. Θέλουν προσοχή καθώς εάν σχολιαστεί μία περιοχή του κώδικα που περιέχει ήδη ένα τέτοιο σχόλιο μέσα της, τότε η περιοχή τερματίζεται στο πρώτο */ που θα συναντηθεί.

Αποδοτική εργασία = Λιγότερα λάθη

Άλλο σημείο προσοχής για τη μείωση των λαθών κατά τη συγγραφή προγραμμάτων είναι το καθαρό μυαλό. Άρα όσο πιο γρήγορα ή ξέγνοιαστα ολοκληρώνεται μια εργασία, τόσο λιγότερα λάθη θα περιέχει.

Η καλή γνώση και χρήση του IDE (CLion) επιτρέπει τη μείωση του χρόνου μπροστά στην οθόνη. Σε αυτό συμβάλει η γνώση των συντομεύσεων του πληκτρολογίου ώστε να μην χάνεται χρόνος στην εναλλαγή ανάμεσα στο πληκτρολόγιο και το ποντίκι. Επίσης «αναγκάζει» τον προγραμματιστή να γνωρίζει περισσότερες από τις λειτουργίες του IDE.

Αν και αλλάζουν κατά τις προτιμήσεις του καθενός, οι συντομεύσεις του CLion υπάρχουν σε αυτή τη διεύθυνση που οδηγεί στο εξής...

IntelliJ IDEA

DEFAULT KEYMAP

Remember these Shortcuts

Smart code completion	Ctrl + Shift + Space
Search everywhere	Double Shift
Show intention actions and quick-fixes	Alt + Enter
Generate code	Alt + Ins
Parameter info	Ctrl + P
Extend selection	Ctrl + W
Shrink selection	Ctrl + Shift + W
Recent files popup	Ctrl + E
Rename	Shift + F6

General

Open corresponding tool window	Alt + #[0-9]
Save all	Ctrl + S
Synchronize	Ctrl + Alt + Y
Toggle maximizing editor	Ctrl + Shift + F12
Inspect current file with current profile	Alt + Shift + I
Quick switch current scheme	Ctrl + BackQuote (`)
Open Settings dialog	Ctrl + Alt + S
Open Project Structure dialog	Ctrl + Alt + Shift + S
Find Action	Ctrl + Shift + A

Debugging

Step over / into	F8 / F7
Smart step into / Step out	Shift + F7 / Shift + F8
Run to cursor	Alt + F9
Evaluate expression	Alt + F8
Resume program	F9
Toggle breakpoint	Ctrl + F8
View breakpoints	Ctrl + Shift + F8

Search / Replace

Search everywhere	Double Shift
Find	Ctrl + F
Find next / previous	F3 / Shift + F3
Replace	Ctrl + R
Find in path	Ctrl + Shift + F
Replace in path	Ctrl + Shift + R
Select next occurrence	Alt + J
Select all occurrences	Ctrl + Alt + Shift + J
Unselect occurrence	Alt + Shift + J

—Productivity Boosters

Editing

Basic code completion	Ctrl + Space
Smart code completion	Ctrl + Shift + Space
Complete statement	Ctrl + Shift + Enter
Parameter info (within method call arguments)	Ctrl + P
Quick documentation lookup	Ctrl + Q
External Doc	Shift + F1
Brief Info	Ctrl +mouse
Show descriptions of error at caret	Ctrl + F1
Generate code...	Alt + Insert
Override methods	Ctrl + O
Implement methods	Ctrl + I
Surround with...	Ctrl + Alt + T
Comment / uncomment with line comment	Ctrl + /
Comment / uncomment with block comment	Ctrl + Shift + /
Extend selection	Ctrl + W
Shrink selection	Ctrl + Shift + W
Context info	Alt + Q
Show intention actions and quick-fixes	Alt + Enter
Reformat code	Ctrl + Alt + L
Optimize imports	Ctrl + Alt + O
Auto-indent line(s)	Ctrl + Alt + I
Indent / unindent selected lines	Tab / Shift + Tab
Cut current line to clipboard	Ctrl+X,Shift+Delete
Copy current line to clipboard	Ctrl+C,Ctrl+Insert
Paste from clipboard	Ctrl+V,Shift+Insert
Paste from recent buffers...	Ctrl+Shift + V
Duplicate current line	Ctrl+D
Delete line at caret	Ctrl+Y
Smart line join	Ctrl+Shift + J
Smart line split	Ctrl+Enter
Start new line	Shift + Enter
Toggle case for word at caret or selected block	Ctrl + Shift + U
Select till code block end / start	Ctrl + Shift +] / [
Delete to word end	Ctrl + Delete
Delete to word start	Ctrl + Backspace
Expand / collapse code block	Ctrl + NumPad+ / -
Expand all	Ctrl+Shift+NumPad+
Collapse all	Ctrl+Shift+NumPad-
Close active editor tab	Ctrl + F4

Refactoring

Copy	F5
Move	F6
Safe Delete	Alt + Delete
Rename	Shift + F6
Refactor this	Ctrl + Alt + Shift + T
Change Signature	Ctrl + F6
Inline	Ctrl + Alt + N
Extract Method	Ctrl + Alt + M
Extract Variable	Ctrl + Alt + V
Extract Field	Ctrl + Alt + F
Extract Constant	Ctrl + Alt + C
Extract Parameter	Ctrl + Alt + P

Navigation

Go to class	Ctrl + N
Go to file	Ctrl + Shift + N
Go to symbol	Ctrl + Alt + Shift + N
Go to next / previous editor tab	Alt + Right/Left
Go back to previous tool window	F12
Go to editor (from tool window)	Esc
Hide active or last active window	Shift + Esc
Go to line	Ctrl+G
Recent files popup	Ctrl+E
Navigate back / forward	Ctrl+Alt + Left/Right
Navigate to last edit location	Ctrl+Shift+Backspace
Select current file or symbol in any view	Alt + F1
Go to declaration	Ctrl + B , Ctrl + Click
Go to implementation(s)	Ctrl + Alt + B
Open quick definition lookup	Ctrl + Shift + I
Go to type declaration	Ctrl + Shift + B
Go to super-method / super-class	Ctrl + U
Go to previous / next method	Alt + Up/Down
Move to code block end / start	Ctrl +]/[
File structure popup	Ctrl + F12
Type hierarchy	Ctrl + H
Method hierarchy	Ctrl + Shift + H
Call hierarchy	Ctrl + Alt + H
Next / previous highlighted error	F2 / Shift + F2
Edit source / View source	F4 / Ctrl + Enter
Show navigation bar	Alt + Home
Toggle bookmark	F11
Toggle bookmark with mnemonic	Ctrl + F11
Go to numbered bookmark	Ctrl + #[0-9]
Show bookmarks	Shift + F11

Compile and Run

Make project	Ctrl + F9
Compile selected file, package or module	Ctrl + Shift + F9
Select configuration and run / debug	Alt + Shift + F10/F9
Run / Debug	Shift + F10 / F9
Run context configuration from editor	Ctrl + Shift + F10

Usage Search

Find usages / Find usages in file	Alt + F7 / Ctrl + F7
Highlight usages in file	Ctrl + Shift + F7
Show usages	Ctrl + Alt + F7

VCS / Local History

Commit project to VCS	Ctrl + K
Update project from VCS	Ctrl + T
Push commits	Ctrl + Shift + K
'VCS' quick popup	Alt + BackQuote (`)

Live Templates

Surround with Live Template	Ctrl + Alt + J
Insert Live Template	Ctrl + J

Ερωτήσεις?

- Διαβάστε τις σημειώσεις, διαβάστε τις διαφάνειες και δείτε τα videos **πριν** ρωτήσετε
- **Συμβουλευτείτε** τη σελίδα ερωταποκρίσεων του μαθήματος

  https://qna.c-programming.allos.gr
- **Στείλτε** τις ερωτήσεις σας πριν και μετά το μάθημα στο

  c-programming@allos.gr

- Εάν έχετε **πρόβλημα** με κάποιο κώδικα στείλτε μαζί τον κώδικα και τα μηνύματα λάθους από το CLion ως κείμενα με copy/paste. Εάν θεωρείτε ότι επιπλέον βοηθά και ένα στιγμιότυπο οθόνης, είναι καλοδεχούμενο.
- Επαναλαμβάνουμε : Μην στείλετε ποτέ κώδικα ως εικόνα μας είναι παντελώς άχρηστος!

Σημαντικά σημεία

Μετά από τη σημερινή διάλεξη θα πρέπει να γνωρίζετε:

- Πώς να καλέσετε μία συνάρτηση
- Πως ορίζω και πως δηλώνω νέες συναρτήσεις
- Τι είναι ο προεπεξεργαστής και που χρησιμοποιείται
- Τι είναι βιβλιοθήκη, ποιες είναι μερικές από τις τυπικές βιβλιοθήκες και που θα βρω τις βασικές συναρτήσεις που παρέχουν
- Πως μπορώ να εργαστώ στο IDE πιο αποδοτικά