

**ΕΡΓΑΣΤΗΡΙΟ 1****Μέρος Α': Εργαλεία – Μεταγλώττιση προγραμμάτων με χρήση GCC compiler**

1. Υπενθύμιση όλων των εργαλείων για σύνδεση από το σπίτι στις μηχανές του εργαστηρίου.
 - a. [vpn](#), [X2Go](#), putty (ssh)
2. Πως θα δουλέψετε στο σπίτι (χωρίς να συνδεθείτε με το εργαστήριο). Επιλογές αν έχετε μηχανή Windows (επιλέξτε ένα από τα πιο κάτω):
 - a. Σύνδεση στις μηχανές του εργαστηρίου μέσω VPN + X2Go. Χρήση gedit ή emacs ή vim ή Eclipse CDT για συγγραφή προγραμμάτων.
 - b. Εγκατάσταση Ubuntu App + VSCode και σύνδεσή τους έτσι ώστε το VSCode να χρησιμοποιεί το GCC compiler του Ubuntu App. Μπορείτε να ακολουθήσετε τον οδηγό <https://code.visualstudio.com/docs/cpp/config-wsl>
 - c. Εγκατάσταση VMWare Player ή Virtuabox για δημιουργία εικονικής μηχανής (VM) με λειτουργικό σύστημα τύπου Unix (π.χ. Ubuntu). Εγκατάσταση GCC compiler μέσα στο λειτουργικό σύστημα. Για editor μπορεί να χρησιμοποιηθεί το gedit ή το VSCode (<https://code.visualstudio.com/docs/setup/linux>).
3. Χρήση Moodle για υποβολή ασκήσεων
4. Επεξεργαστές κειμένου σε περιβάλλον Linux – *emacs* (υπάρχει υλικό στο αρχείο [lab1.files.zip](#)), *gedit*
5. Δομή ενός προγράμματος C
Ένα πρόγραμμα C έχει την ακόλουθη δομή:

```
[εντολές προς τον προ-επεξεργαστή]  
[δήλωση καθολικών μεταβλητών]  
[συναρτήσεις]
```

```
main() {  
    δήλωση τοπικών μεταβλητών  
    λοιπές εντολές  
}
```

6. Μεταγλώττιση ενός προγράμματος C
 - a. Κατεβάστε το αρχείο [lab1.files.zip](#) και αποσυμπίστε.
 - b. Μεταγλωττίστε το πρόγραμμα `CtoF.c` με την ακόλουθη εντολή
`gcc CtoF.c`
 - c. Εκτελέστε το πρόγραμμα με την εντολή
`./a.out`
 - d. Μεταγλωττίστε ξανά το πρόγραμμα `CtoF.c` με την ακόλουθη εντολή
`gcc CtoF.c -o CtoF`
 - e. Εκτελέστε το πρόγραμμα με την εντολή
`./ CtoF`
 - f. Μπορείτε να χρησιμοποιήσετε και άλλες εντολές όπως:
`gcc CtoF.c -Wall -o CtoF`
Σημείωση: για να χρησιμοποιήσετε την βιβλιοθήκη **math.h** προσθέστε το `-lm`
`gcc add.c -o add -lm`
 - g. Μεταγλωττίστε το πρόγραμμα `sample-lab1.c` χρησιμοποιώντας διάφορα ορίσματα του μεταγλωττιστή gcc (`-E`, `-S`, `-c`, `-o`). Δείτε διαφάνεια 1-28 από τις διαλέξεις.

**Μέρος Β': Λογικοί και αριθμητικοί τελεστές**

1. Μελετήστε τα πιο κάτω 3 προγράμματα προσεκτικά.

- Γράψτε στο χαρτί τα πιθανά αποτελέσματα των αριθμητικών εκφράσεων και της εκτύπωσής τους.
- Μεταγλωττίστε τα προγράμματα και συγκρίνετε τα αποτελέσματα εκτέλεσής τους με τα δικά σας.

```
/** @file operators1.c
 * @brief Operator precedence and data types. Use of formatting.
 *
 * This program performs various arithmetic operations. Its purpose is
 * to demonstrate the importance and priority of operators, data types,
 * and the use of formatting.
 *
 * @author Pavlos Antoniou
 * @bug No known bugs.
 */

#include <stdio.h>

int main(){
    /* Variable declaration */
    int int1, int2, int3, int4;
    float float1, float2, float3, float4;

    /* Arithmetic operations */
    int1 = 2;
    int2 = 3;
    int3 = 2/3;
    int4 = 3.0/2;
    float1 = 2/3;
    float2 = 2.0/3;
    float3 = int1 * int2 + (float1 + 1) / float2;
    float4 = 2.0 * 3 + 4 - 2 * 4 * 2.0 * 1.5;

    /* Print results */
    printf("\n Ta apotelesmata einai:\n");
    printf("\t int3 = %d\n", int3);
    printf("\t int4 = %d\n", int4);
    printf("\t int4 (f) = %f\n", int4);
    printf("\t float1 = %f\n", float1);
    printf("\t float2 = %f\n", float2);
    printf("\t float2 (e) = %e\n", float2);
    printf("\t float2 (d) = %d\n", float2);
    printf("\t float2/float1 = %f\n", float2/float1);
    printf("\t float3 = %f\n", float3);
    printf("\t float2 + float3 = %f\n", float2 + float3);
    printf("\t float2 + float3 = %5.2f\n", float2 + float3);
    printf("\t float4 = %f\n\n", float4);
    return 0;
}
```



```
/** @file operators2.c
 * @brief Operator precedence and data types. Use of formatting.
 *
 * This program performs various logical operations. Its purpose is
 * to demonstrate the importance and priority of operators, data types,
 * and the use of formatting.
 *
 * @author Pavlos Antoniou
 * @bug No known bugs.
 */

#include <stdio.h>

int main() {

    int a=1, b=1, c=0, d=0;

    printf("a = %d, b = %d, c = %d, d = %d.\n", a, b, c, d);

    printf("a || b && c || d = %d.\n", a || b && c || d);
    printf("(a||b) && c || d = %d.\n", (a||b) && c || d);
    printf("a || (b&&c) || d = %d.\n", a || (b&&c) || d);
    printf("a || b && (c||d) = %d.\n", a || b && (c||d));

    return 0;
}

/** @file operators3.c
 * @brief Operator precedence and data types. Use of formatting.
 *
 * This program performs various logical and arithmetic operations.
 * Its purpose is to demonstrate the importance and priority of operators,
 * data types, and the use of formatting.
 *
 * @author Pavlos Antoniou
 * @bug No known bugs.
 */

#include <stdio.h>

int main() {
    printf("Bonjour.\n");

    printf("-4-3*2-1 = %d\n", -4-3*2-1);

    printf("-5/2 = %d\n", -5/2);
    printf("2*-5/2 = %d\n", 2*-5/2);
    printf("4-5-3==2*-5/2+1 = %d\n", 4-5-3==2*-5/2+1);

    printf("1+3*8%3 = %d\n", 1+3*8%3);

    int n1 = 20, x1;
    x1 = n1++; printf("x1 = %d\n", x1);

    int n2 = 20, x2;
```



```
x2 = ++n2; printf("x2 = %d\n", x2);

printf("32&&2.3 = %d\n", 32&&2.3);

printf("!65.34 = %d\n", !65.34);

printf("0||!(32>12) = %d\n", 0||!(32>12));

int X1, A=5, B=10, C=1;
X1 = ((2*A+3)*B+4);
printf("X1 = %d\n", X1);

int X2=3, Y=4;
X2 *= Y+1;
printf("X2 = %d\n", X2);

printf("Au revoir.\n");
return (0);
}
```

Μέρος Γ': Συγγραφή απλών προγραμμάτων

Σειριακά προγράμματα

1. Γράψτε ένα πρόγραμμα που μετατρέπει *Fahrenheit* σε *Celsius*.
 $Celsius = 5/9 * (Fahrenheit - 32)$
2. Γράψτε ένα πρόγραμμα που διαβάζει δυο ακέραιους και τυπώνει το άθροισμα και το γινόμενο τους.
3. Γράψτε ένα πρόγραμμα που διαβάζει έναν ακέραιο από τον χρήστη και τυπώνει στην οθόνη εάν είναι μονός ή ζυγός αριθμός.
4. Γράψτε ένα πρόγραμμα που ζητά από τον χρήστη δυο ακέραιους αριθμούς, και τυπώνει το πρόσημο της άθροισης των δυο αριθμών, χωρίς να γίνει η άθροιση.

Προγράμματα με χρήση συνθήκης: if, else

5. Γράψτε ένα πρόγραμμα **calculator.c** το οποίο:
διαβάζει δυο πραγματικούς αριθμούς και ένα συντελεστή στην μορφή:
 $number1 \ operator \ number2$
και ανάλογα με τον συντελεστή κάνει την αριθμητική πράξη και τυπώνει το αποτέλεσμα στην μορφή:
 $number1 \ operator \ number2 = result$
όπου *operator* είναι ένας εκ των +, -, *, /
Να γίνεται έλεγχος για λάθος εισαγωγή συντελεστή και να τυπώνεται κατάλληλο μήνυμα σε περίπτωση σφάλματος.
6. Γράψτε ένα πρόγραμμα που κάνει την προσομοίωση της διαίρεσης δυο θετικών ακέραιων που εισάγονται από τον χρήστη. Το πρόγραμμα, διαιρεί το μεγαλύτερο με τον μικρότερο χωρίς να χρησιμοποιεί τον τελεστή / , και τυπώνει το πηλίκο και το υπόλοιπο της διαίρεσης.