

Τίτλος Εργασίας : “ΤΕΤΑΡΤΗ 08:00 - 10:00 ΕΡΓΑΣΙΑ 1 “

Αριθμός : 1

Στέφανος Στεφάνου

Τμήμα: Α3

Ερώτημα 1.1) “Τί είναι πρόγραμμα; “

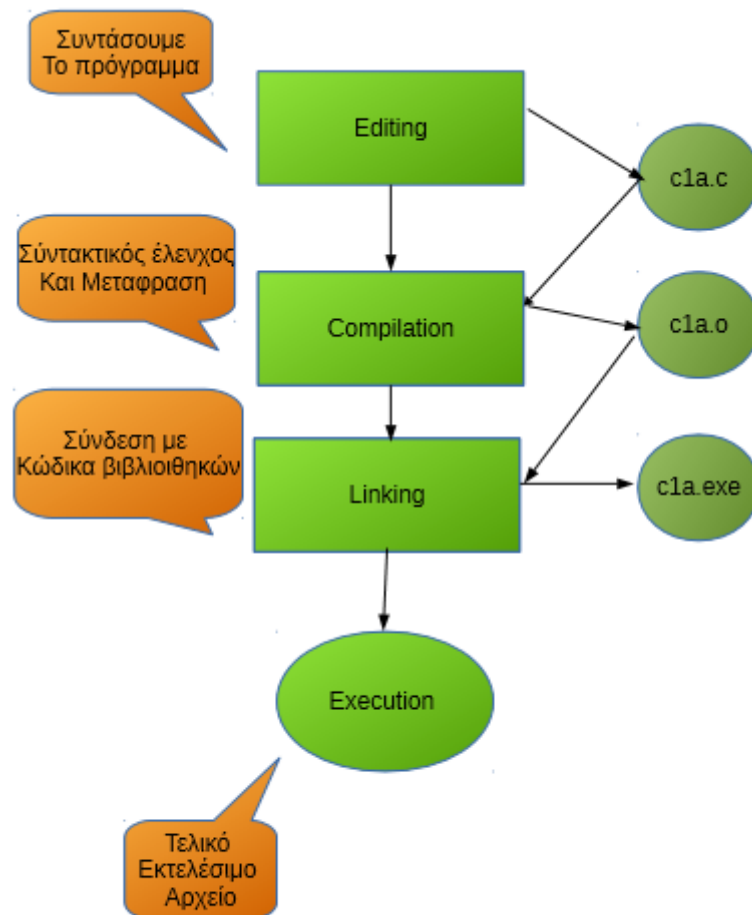
Απάντηση : Πρόγραμμα είναι μία συγκεκριμένη ακολουθία εντολών , τις οποίες πρέπει να εκτελέσει ο υπολογιστής μια προς μια , προκειμένου να παράγει το επιθυμητό , προς τον χρήστη , αποτέλεσμα

Ερώτημα 1.2)“Ποια είναι τα βήματα ανάπτυξης και εκτέλεσης ενός C προγράμματος;”

Τα βασικά βήματα ανάπτυξης και εκτέλεσης ενός προγράμματος σε C είναι τα εξής ...

1. Γράφουμε το πρόγραμμα, την σειρά των εντολών σε έναν οποιοδήποτε κειμενογράφο , και αποθηκεύουμε το αρχείο με επέκταση “.c”
2. Μετά ξεκινάμε την μεταγλώτιση , το αρχείο περνάει πρώτα από τον μεταγλωτιστή , ο μεταγλωτιστής , αφού ελέγχει το πρόγραμμα για συντακτικά λάθη , προχωρά στην μετάφραση του προγράμματος σε γλώσσα μηχανής , η οποία δεν είναι άμεσα εκτελέσιμη από τον ίδιο τον υπολογιστή , διότι ακόμα δεν έχει συνδεθεί με τις κατάλληλες βιβλιοθήκες . Ο κώδικας σε αυτήν την κατάσταση λέγεται “Αντικείμενος κώδικας” , και είναι μια ημιτελή τελική έκδοση του εκτελέσιμου κώδικα μηχανής . Ο αντικείμενος κώδικας έχει κατάληξη αρχείου “.o” σε linux συστήματα ή “.obj” σε συστήματα windows
3. Ο αντικείμενος κώδικας περνάει από το στάδιο της σύνδεσης(Linking) με τον κώδικα των βιβλιοθηκών. ύστερα από το στάδιο αυτό . Ο παραγόμενος κώδικας είναι εκτελέσιμος , κάτω από περιβάλλον windows τα αρχεία που περιέχουν εκτελέσιμο κώδικα έχουν κατάληξη “.exe”

Το παρακάτω σχεδιάγραμμα απεικονίζει την διαδικασία σύνταξης και εκτέλεσης των εντολών



Ερώτημα 1.3) "Ποιες είναι οι γενικές κατηγορίες λαθών των προγραμμάτων; "
Οι γενικές κατηγορίες λαθών των προγραμμάτων είναι οι εξής

- Συντακτικά λάθη(Syntax Errors)

Τα Συντακτικά λάθη προκύπτουν , όταν δεν ακολουθούνται συντακτικοί κανόνες της εκάστοτε γλώσσας προγραμματισμού. Εντοπίζονται και καταγράφονται από τον μεταγλωττιστή (Compiler) της γλώσσας προγραμματισμού

- Λογικά λάθη (Logic Errors)

Τα λογικά λάθη προκύπτουν , όταν το πρόγραμμα δεν λειτουργεί με τον επιθυμητό τρόπο , δηλαδή όταν υπάρχει λάθος στον αλγόριθμο πίσω από το πρόγραμμα , και όχι στο πρόγραμμα καθ' αυτό .

- Σφάλματα κατά τον χρόνο εκτέλεσης(Runtime Errors)

Προκύπτουν κατά τον χρόνο εκτέλεσης , όταν μια ή περισσότερες εντολές δεν είναι εφικτό να εκτελεστούν , ως συνέπεια το πρόγραμμα "κρεμάει" ("κρассαρει")

Ερώτημα 2)

Κατεβάστε από το eClass το αρχείο "c1a.c". Το αρχείο περιέχει ένα πρόγραμμα το οποίο έχει ως στόχο την εκτέλεση των τεσσάρων βασικών πράξεων, σε δύο ακέραιους αριθμούς που εισάγονται από το πληκτρολόγιο. Δοκιμάστε να το μεταγλωττίσετε. Καταγράψτε, εξηγήστε και διορθώστε τα συντακτικά λάθη του προγράμματος "

Απάντηση ερωτήματος 2)

Μεταγλωττίζοντας τον αρχικό πηγαίο κώδικα , ο οποίος παρατίθεται παρακάτω , προκύψανε τα εξής λάθη

*Σημείωση 01 , το πρόγραμμα μεταφράστηκε πάνω σε kubuntu 16.04 "Xenial Xerus" 64bit με μεταγλωττιστή gcc 6.2 , επίσης χρησιμοποιήθηκε ο Sublime Text Editor (εκδοση 2.0)

*Σημείωση 02 : λόγω ότι η ANSI C υποστηρίζει μόνο το πρότυπο ASCII για αναπαράσταση και χειρισμό συμβολοσειρών , οι προς εκτύπωση συμβολοσειρές μετατραπήκαν στην αγγλική γλώσσα

*Σημείωση 03 :Εναλλακτικά , θα μπορούσε να χρησιμοποιηθεί η ελληνική κωδικοποίηση με χρήση της εντολής system("chcp 1252") ή system("chcp 65000"), η συνάρτηση system() υπάρχει μέσα στο αρχείο stdlib.h , οπότε να σημειωθεί ότι πρέπει να συμπεριλάβουμε την εντολή #include <stdlib.h>
#include <stdio.h>

```

#include <stdio.h>
int main(int argc, int **argv)
{
    int A, B;
    int C, D, E, F;
    printf ("Basic nummeric operations with integers\n");
    printf ("=====\\n\\n");
    printf ("Give first number : ");
    scanf ("%d",&A);
    printf ("Give second number : ");
    scanf ("%d", &B);
    C = A + B;
    D = A - B;
    E = A * B;
    F = A / B;
    printf ("Add : %d\\n", C);
    printf ("Revision : %d\\n", D);
    printf ("Multiply : %d\\n", E);
    printf ("Division : %d\\n", F);
    return 0;
}

```

Λάθος 01) Γραμμή 10 → Λείπει ο τερματικός χαρακτήρας συμβολοσειρών

Μύνημα λάθους -> <<warning: missing terminating " character>>

Εξήγηση → Κάθε συμβολοσειρα στην γλώσσα C ξεκινάει με διπλά αγγλικά εισαγωγικά (χαρακτήρας ") και τερματίζει επίσης με διπλά εισαγωγικά , όπως παρατηρούμε στην εντολή της γραμμής 11 , το διπλό εισαγωγικό τερματισμού της συμβολοσειράς λείπει , και έτσι ο Compiler της γλώσσας παρουσιάζει σφάλμα

Έτσι , η διορθωμένη έκδοση της γραμμής 10 θα ήταν...

```
10    printf ("Give first number : ");
```

Λάθος 02) Γραμμη 16 → η μεταβλητη c δεν εχει δηλωθεί στην προκείμενη εμβέλεια...

Μύνημα λάθους -> <<error: 'c' was not declared in this scope>>

Εξήγηση -> Αυτό συμβαίνει , διότι στην γραμμή 12 , όταν είχαμε δηλώσει την μεταβλητη C την είχαμε δηλώσει με κεφαλαίο χαρακτήρα C και όχι με μικρό . Εδώ ας τονίσουμε ότι η γλώσσα προγραμματισμού C είναι CaseSensitive . Δηλαδή θεωρεί ένα αναγνωριστικό (Identifier) ίδιο από ένα άλλο , μόνο αν έχει γραφτεί με τον ίδιο ακριβώς τρόπο , χωρίς διαφορές σε κεφαλαία-μικρά γράμματα! Αρα οι τα κάτωθι αναγνωριστικά δεν είναι ίδια

- aa
- Aa
- aA
- AA

..Έτσι η διορθωμένη έκδοση της εντολής 16 είναι ...

```
16    printf ("Add : %d\n", C);
```

Λάθος 03) Γραμμή 19 → Η μεταβλητή E έχει γραφτεί με χαρακτήρα που δεν υπάρχει στο πρότυπο ASCII

Μύνημα λάθους -> <<error: stray '\316' in program>>

Εξήγηση -> Το λάθος αυτό, προέκυψε, διότι η μεταβλητή E έχει γραφτεί με ελληνικό χαρακτήρα “Ε” και όχι με αγγλικό χαρακτήρα “E”. Εδώ πρέπει να τονίσουμε ότι τα αναγνωριστικά της C, καθώς και όλες οι εντολές, μπορούν να γραφτούν μόνο με λατινικούς χαρακτήρες, όπως αυτούς ορίζει το πρότυπο ASCII

...Έτσι η διορθωμένη έκδοση της γραμμής 19 θα είναι...

```
18    printf ("Multiply : %d\n", E);
```

Λάθος 04) Γραμμή 19 -> Το λάθος αυτό, προέκυψε διότι έχει γίνει λάθος στο όνομα της συνάρτησης printf() το οποίο έχει γραφτεί prantf(). Το συγκεκριμένο λάθος δεν είναι εύκολο να εντοπιστεί, διότι συνέβη κατά το στάδιο του linking, δηλαδή της σύνδεσης του αντικειμένου κώδικα με την βιβλιοθήκη των ήδη δηλωμένων συναρτήσεων. Αυτό συνεβή επειδή σαν σύνταξη, η εντολή της γραμμής 19 είναι απόλυτα σωστή, ο συντακτικός αναλυτής ελένχει μόνο την σύνταξη των εντολών χωρίς να ελένχει αν αυτές είναι εγκύρες. Έτσι η γραμμή 19 πέρασε τον συντακτικό έλεγχο, και καθώς ο συνδέτης εψαχνέ τις βιβλιοθήκες της γλώσσας, δεν βρήκε κάποια συνάρτηση prantf() και έτσι προέκυψε το λάθος.

Σε λάθη τετοιού τύπου, λόγω ότι στο σημείο αυτό ο κώδικας έχει μετατραπεί σε ημιτελή κώδικα μηχανής (δηλαδή κώδικας που δεν μπορεί να τρέξει άμεσα, λόγω ότι δεν έχει ακόμα συνδεθεί) δεν μπορεί να εντοπιστεί η γραμμή του σφάλματος στον κωδικά μας.

Ωστόσο, μερικά ολοκληρωμένα προγραμματιστικά περιβάλλοντα αναγνωρίζουν και εντοπίζουν και τετοια είδη λαθών, δοκιμάζοντας στο περιβάλλον Clion της Brainstorm κάτω από Windows 10 64bit με μεταγλωτιστή τον MinGW. Μπορούμε να πάρουμε το εξής μήνημα λάθους

Μύνημα λάθους -> <<error: 'prantf' was not declared in this scope>>

...Έτσι η διορθωμένη έκδοση της γραμμής 19 θα είναι....

```
19)printf ("Division : %d\n", F);
```

Ερώτημα 03) “Εκτελέστε το πρόγραμμα. Καταγράψτε την έξοδό του για διάφορες τιμές εισόδου και τις όποιες παρατηρήσεις σας.”

Απάντηση ερωτήματος 03) Ας τρέξουμε το πρόγραμμα με διάφορες τιμές για να καταγράψουμε την έξοδο του και να εντοπίσουμε τυχόν παραλείψεις του....

1) Σετ τιμών : 5,10

Ανταπόκριση προγράμματος

Eisagete ton prwto arithmo:5

Eisagete ton deuthro aritmo :10

```
Add      : 15
Revision   : -5
Multiply   : 50
Division   : 0
```

Σχόλια : το πρόγραμμα για ένα απλό σετ τιμών ανταποκρίθηκε άριστα.

2) Σετ τιμών : 5 0

Ανταπόκριση προγράμματος ...

Give first number:5

Give second number :0

Process finished with exit code 255

Σχόλια : το συγκεκριμένο σετ τιμών , κρύβει μια παγίδα , διότι δεν υπάρχει κανείς έλεγχος για τις τιμές που εισάγονται , έτσι ο υπολογιστής , προσπαθεί να κάνει , μάταια...."διαίρεση με το μηδέν" . Έτσι το πρόγραμμα παρουσιάζει σφάλμα κατά τον χρόνο εκτέλεσης (Runtime Error) και διακόπτει την λειτουργία του απροσδόκητα.....

3) Σετ τιμών : -10 -20 (1ό scanf) απλό enter στο επόμενο scanf

Ανταπόκριση προγράμματος

```
Give first number:-10 -20
Give second number aritmo :
```

```
Add      : -30
Revision   : 10
Multiply   : 200
Division   : 0
```

Σχόλια :

Το πρόγραμμα τερμάτισε επιτυχώς , παρόλο που βάλαμε 2 τιμές στο πρώτο scanf() και καμία τιμή στο δεύτερο scanf(). Αυτό συμβαίνει λόγω του τρόπου που χειρίζεται η scanf() τα δεδομένα εισόδου (stdin) . Η scanf() θεωρεί το κενό χαρακτήρα (" ") ως διαχωριστικό μεταξύ των δεδομένων , και όταν η scanf διαβάζει παραπάνω από τα δεδομένα που αντιστοιχούν στις μεταβλητές δεξιά της , τις καταχωρεί σε μια προσωρινή διεύθυνση (buffer) για επόμενες κλήσεις της scanf() . Έτσι την πρώτη φορά διαβάστηκε το -10 και καταχωρήθηκε στην μεταβλητή A και το -20 καταχωρήθηκε στον buffer. Μόλις η scanf ξανακαλέστηκε τότε αυτομάτως εναπόθεσε την τιμή -20 στην μεταβλητή B.

4) Σετ τιμών a , b

Ανταπόκριση προγράμματος

Give first number:a

Give second number :b

Add : 10623439
Revision : -2221401
Multiply : -276107492
Division : 0

Σχόλια : παρόλο που εισάγαμε χαρακτήρες , το πρόγραμμα επεστρεψε κανονικά αποτελέσματα . Αυτό συμβαίνει διότι οι χαρακτήρες , κωδικοποιούνται στην μνημη ως απλοί ακειριοι αριθμοί . Έτσι , παρόλο λαθός , το πρόγραμμα ανταποκρίνεται κανονικά.

Εδώ αξίζει να αναφέρουμε οτι ακόμα και ο τύπος δεδομένων "char " είναι ένας ειδικός τύπος ακειριου αριθμού

Ερώτημα 4) " Εκτελέστε το πρόγραμμα δίδοντας ως είσοδο μεγάλους ακέραιους αριθμούς. Τι παρατηρείτε; Που οφείλεται το αποτέλεσμα; "

Απάντηση ερωτήματος 04)
Σετ τιμών 999999999999 556
Ανταπόκριση προγράμματος

Give first number:999999999999

Give second number :556

Athroisma : -727379413
Ypoloipo : -727380525
Ginomeno : -696336940
Diairesh : -1308237

Σχόλια: το πρόγραμμα για ενα μεγάλο σετ αριθμών παρουσιαζει λάθος αποτελεσματα, το λάθος αυτό οφείλεται στον τρόπο που η C χειριζεται την μνημη και τους ακειριους αριθμους. Οταν ενας αριθμός στην μνήμη ξεπεράσει το επιτρεπόμενο όριο, τότε αυτόματα ο αριθμός επιστρέφει στην αρχή του ευρους αριθμων που υποστηρίζει η C . έτσι αν είχαμε υποθετικά εύρος τιμών απο -10 ως +10 , τότε το αποτέλεσμα της πράξης 9+3 θα μας εκανε -9

Ερώτημα 5) "Προτείνετε, σε φυσική γλώσσα, βελτιώσεις που θα έπρεπε να γίνουν στο πρόγραμμα ώστε να είναι κατάλληλο για επαγγελματική χρήση (θεωρώντας πως κάποιος θα είχε ανάγκη, σήμερα, για ένα τέτοιο πρόγραμμα"

Απάντηση ερωτήματος 05)
Το παραπάνω πρόγραμμα , θα μπορούσε να πάρει πολλές βελτιώσεις . Χαρακτηριστικα μερικες απο αυτες είναι....

1. Θα μπορούσε να γίνεται έλεγχος για μη=μη μηδενικη τιμή στην δεύτερη scanf μονο (ας θυμηθούμε οτι 0/5 ορίζεται και κάνει 0)
2. Θα μπορούσε να γίνεται έλεγχος ,ετσι ώστε να δίνεται μόνο μια τιμή , ανα κλήση της scanf() , για να αποφύγουμε λάθη οπως αυτά του Σετ τιμων #2
3. Θα μπορούσε να γίνεται έλεγχος για την εισαγωγή μόνο αριθμητικών στοιχείων , για να αποφύγουμε λάθη , όπως αυτα του Σέτ τιμών #4

4. θα μπορούσαμε να υπολογίζουμε ακόμα και το υπολοιπό διαίρεσης , η ακόμα και να υποστηρίξουμε πράξεις με αριθμούς κινητής υποδιαστολής