

Ινστιτούτο Επαγγελματικής Κατάρτισης ΑΛΦΑ



C Programming

Π22Β-ΑΛΓΟΡΙΘΜΙΚΗ ΚΑΙ ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ /
ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Γιαννουλάτος Λάμπρος

Α' Εξάμηνο
2022-2023

ΠΕΡΙΕΧΟΜΕΝΑ

01 Το ζητούμενο
σελίδες: 1

02 Ο αλγόριθμος
σελίδες: 2-3

03 Ο κώδικας
σελίδες: 4-9

Το ζητούμενο

Κάθε ανάγκη για τη δημιουργία ενός προγράμματος ή ακολουθίας εντολών αποσκοπεί στο να λυθεί ένα πρόβλημα που αντιμετωπίζουμε. Ο καλύτερος τρόπος για να λύσουμε ένα πρόβλημα που αντιμετωπίζουμε είναι πρώτα να το κατανοήσουμε.

Ακολουθεί η εκφώνηση του προβλήματος:

Να δημιουργηθεί ένα πρόγραμμα με τη γλώσσα C με διάφορες μαθηματικές δυνατότητες. Συγκεκριμένα, το πρόγραμμα θα έχει τις εξής δυνατότητες:

1) Θα ανοίγει το πρόγραμμα, Θα ζητάει δύο αριθμούς και στη συνέχεια θα σου εμφανίζει να εισάγεις κάτι για να επιλέξεις αν θέλεις να κάνεις πρόσθεση, αφαίρεση ή πολλαπλασιασμό και στο τέλος θα σου εμφανίζει το αντίστοιχο αποτέλεσμα.

2) Αφού έχει φτιαχτεί το πρώτο βήμα θα πρέπει το πρόγραμμα να προσαρμοστεί όπου να μην κλείνει όταν εμφανίζει το αποτέλεσμα μία πράξης αλλά να ξαναεμφανίζει το μενού του προγράμματος.

Σε αυτό το βήμα θα πρέπει να προστεθεί με δομή μενού και η δυνατότητα κλεισίματος του προγράμματος ή υλοποίησης κάποιας πράξης.

π.χ. Εισήγαγε 1 για πράξη , Εισήγαγε 2 για να κλείσει το πρόγραμμα.

3) Αφού έχουν υλοποιηθεί τα 2 προηγούμενα βήματα θα προστεθεί στο μενού και ένα 3ο κουμπί όπου θα ζητάει έναν αριθμό και θα σου εμφανίζει αν είναι μονός ή ζυγός αριθμός.

Ο αλγόριθμος

Όπως είπαμε και προηγουμένως είναι σημαντικό πριν ξεκινήσουμε να γράφουμε κώδικα να κατανοήσουμε το ζητούμενο. Το ζητούμενο, είναι ένα πρόβλημα προς επίλυση. Οπότε αφού διαβάσουμε αρκετές φορές την εκφώνηση με προσοχή, το επόμενο βήμα είναι να ξεκινήσουμε να σκεφτόμαστε το πως θα φτάσουμε στην επίλυση.

Το πρώτο βήμα λοιπόν, είναι να καταγράψουμε όλα εκείνα τα βήματα που θα χρειαστούμε για να φτάσουμε στην επίλυση. Έτσι λοιπόν, θα δημιουργήσουμε μία σειρά από λογικές εντολές - σκέψεις που στη συνέχεια θα μας βοηθήσουν να γράψουμε τον κώδικα μας. Το σύνολο αυτών των εντολών - σκέψεων ονομάζεται Αλγόριθμος. Ένα απλό και κοινό σε όλους μας παράδειγμα ενός αλγορίθμου είναι μία μαγειρική συνταγή. Μία συνταγή έχει όλα εκείνα τα απαραίτητα βήματα για να φτάσουμε στο αποτέλεσμα που επιθυμούμε.

Ο αλγόριθμος μπορεί να αποτυπωθεί με πολλούς τρόπους. Μπορεί να γραφτεί απλά με τα βήματα του, το ένα μετά το άλλο. Μπορεί να γραφτεί με την μορφή της Ψευδογλώσσας ή ακόμα και με την μορφή διαγράμματος ροής.

2

Για την συγκεκριμένη εργασία θα παρουσιάσουμε δύο τρόπους με τους οποίους θα λύσουμε το ζητούμενο, τον αλγόριθμο το διάγραμμα ροής.

Στη συνέχεια ακολουθεί ο αλγόριθμος για την εργασία μας:

Αλγόριθμος με τη μορφή βημάτων:

Βήμα 1: Αρχή (δήλωση βιβλιοθηκών, μεταβλητών)

Βήμα 2: Εισαγωγή δύο ακέραιων αριθμών

Βήμα 3: Εμφάνιση μενού

Βήμα 4: Επιλογή χρήστη από το μενού

Βήμα 5: Έλεγχος επιλογής

Βήμα 6: Πραγματοποίηση λειτουργίας

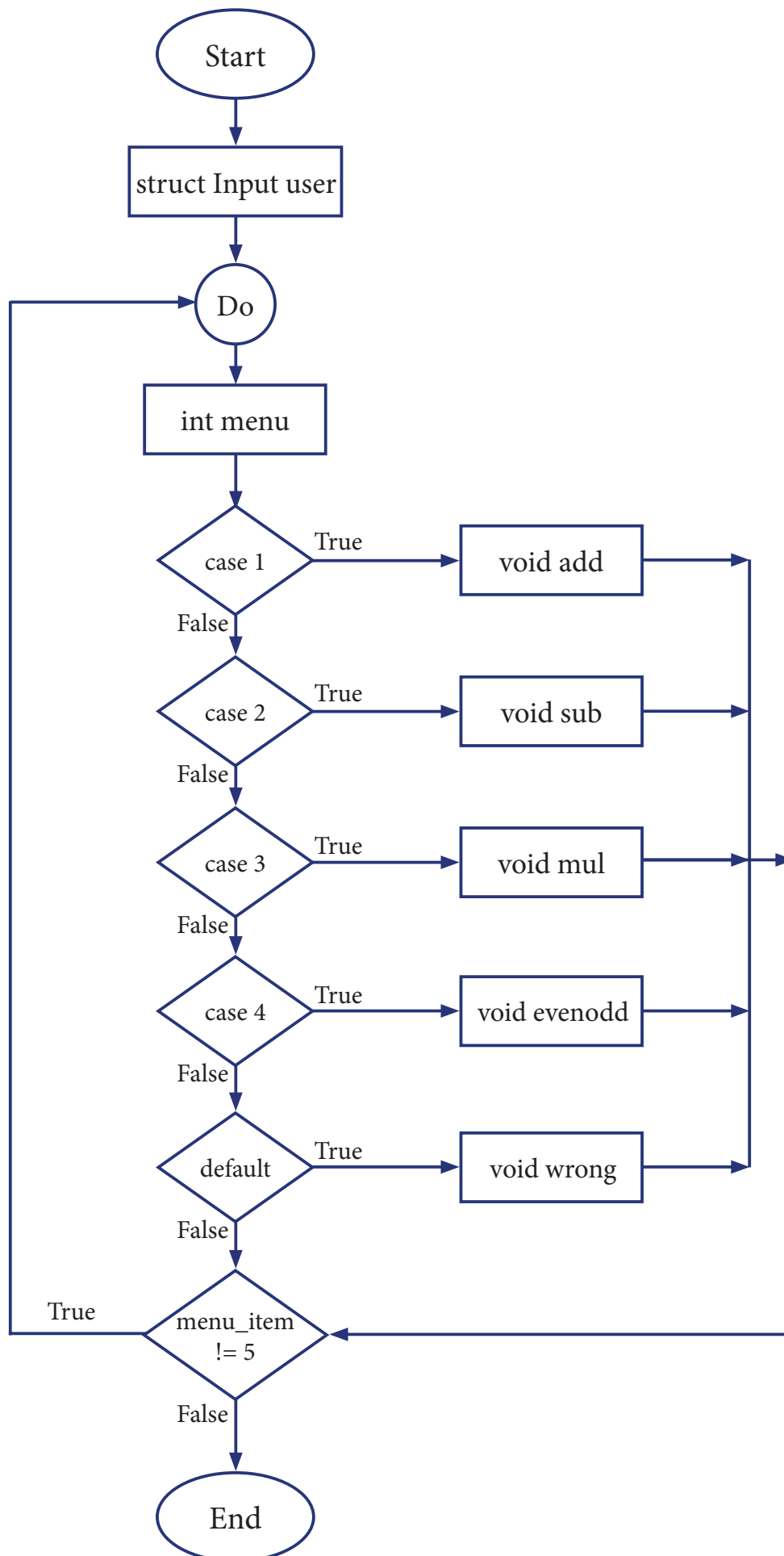
Βήμα 7: Επιστροφή στο Βήμα 3 (Δομή επανάληψης)

Βήμα 8: Τέλος

Ο αλγόριθμός μας, περιέχει ένα ένα τα βήματα με τις λογικές σκέψεις του προγράμματος μας.

Ας δούμε και έναν άλλον τρόπο με τον οποίο μπορούμε να απεικονίσουμε τις σκέψεις μας και εν τέλει τον αλγόριθμο για το πρόβλημα μας.

Ακολουθεί το διάγραμμα ροής:



Ακολουθεί ο κώδικας:

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct Input{
5      int number_1;
6      int number_2;
7  };
8
9  struct Input user(){
10     struct Input input;
11     printf("Enter the first Integer number:");
12     scanf("%d",&input.number_1);
13     printf("Enter the second Integer number:");
14     scanf("%d",&input.number_2);
15     return input;
16 }
17
18 int menu(){
19     int choice = 0;
20     printf("\n");
21     printf("Make your selection please\n");
22     printf("=====\\n");
23     printf("|| 1. Addition +                ||\\n");
24     printf("|| 2. Substraction -              ||\\n");
25     printf("|| 3. Multiplication *             ||\\n");
26     printf("|| 4. Even or Odd                  ||\\n");
27     printf("|| 5. Exit                          ||\\n");
28     printf("=====\\n");
29     scanf("%d",&choice);
30     return choice;
31 }
32
33 void add(int num_1,int num_2){
34     int sum= num_1+num_2;
35     printf("\n");
36     printf(" ~ Addition Mode ~\\n");
37     printf("The Addition of %d and %d is: %d\\n",num_1,num_2,sum);
38     printf("\n");
39 }
40
41 void sub(int num_1,int num_2){
42     int sub= num_1-num_2;
43     printf("\n");
44     printf(" ~ Substraction Mode ~\\n");
45     printf("The Substraction of %d and %d is: %d\\n",num_1,num_2,sub);
46     printf("\n");
47 }
48
49 void mul(int num_1,int num_2){
50     int mul= num_1*num_2;
51     printf("\n");

```

```

52     printf(" ~ Multiplication Mode ~\n");
53     printf("The Multiplication of %d and %d is: %d\n",num_1,num_2,mul);
54     printf("\n");
55 }
56
57 void evenodd(){
58     int a = 0;
59     printf("\n");
60     printf(" ~ Even or Odd Mode ~\n");
61     printf("Enter an integer number\n");
62     scanf("%d",&a);
63     if (a%2 == 0) //modulus operator
64         printf("The given number %d is even\n",a);
65     else
66         printf("The given number %d is odd\n",a);
67     printf("\n");
68 }
69
70 int wrong(int number){
71     printf("\n");
72     if (number != 5)
73         printf("Wrong selection number\n");
74 }
75
76 int main() {
77     struct Input input = user();
78     int menu_item = 0;
79     do
80     {
81         menu_item = menu();
82         switch(menu_item)
83         {
84             case 1:
85                 add(input.number_1,input.number_2);
86                 break;
87             case 2:
88                 sub(input.number_1,input.number_2);
89                 break;
90             case 3:
91                 mul(input.number_1,input.number_2);
92                 break;
93             case 4:
94                 evenodd();
95                 break;
96             default:
97                 wrong(menu_item);
98                 break;
99         }
100     } while (menu_item != 5);
101     printf("Program is closing...");
102     return 0;
103 }

```

Στη συνέχεια θα εξηγήσουμε τη λειτουργία του κώδικα και κάποια σημεία που αξίζουν να σχολιαστούν.

Η βασική λειτουργία του προγράμματος μας αφορά ένα μενού το οποίο θα εμφανίζεται πριν και μετά από κάθε λειτουργία. Επίσης το μενού θα κλείνει μόνο αν ο χρήστης δώσει τον αριθμό 5. Η εμφάνιση του μενού γίνεται από μία συνάρτηση την οποία την έχουμε ονομάσει menu(). Στη συνέχεια παραθέτουμε την συνάρτηση καθώς και την κλήση της από το main πρόγραμμα.

```
int menu(){
    int choice = 0;
    printf("\n");
    printf("Make your selection please\n");
    printf("=====\n");
    printf("|| 1. Addition +           ||\n");
    printf("|| 2. Subtraction -        ||\n");
    printf("|| 3. Multiplication *      ||\n");
    printf("|| 4. Even or Odd           ||\n");
    printf("|| 5. Exit                  ||\n");
    printf("=====\n");
    scanf("%d",&choice);
    return choice;
}
```

Η κλήση γίνεται από την main και στη θέση της επιστρέφεται η επιλογή του χρήστη για το μενού.

```
int main() {
    struct Input input = user();
    int menu_item = 0;
    do
    {
        menu_item = menu();
        switch(menu_item)
        {
            .
            .
            .
            .
        }
    } while (menu_item != 5);
    printf("Program is closing...");
    return 0;
}
```


Στη συνέχεια θα δούμε αναλυτικά το τι συμβαίνει στο πρόγραμμα μας μετά την επιλογή του χρήστη για το μενού και πως γίνονται οι πράξεις. Προηγουμένως είδαμε πως περνάμε την επιλογή του χρήστη από την συνάρτηση μας στην `main`.

Παίρνουμε την επιλογή του χρήστη και την περνάμε από έναν έλεγχο μέσα σε μια `switch`. Μέσα στην `switch` ανάλογα με το ποιά είναι η επιλογή του χρήστη θα γίνει και η αντίστοιχη κλήση της συνάρτησης.

```
do
{
    menu_item = menu();
    switch(menu_item)
    {
        case 1:
            add(input.number_1,input.number_2);
            break;
        case 2:
            sub(input.number_1,input.number_2);
            break;
        case 3:
            mul(input.number_1,input.number_2);
            break;
        case 4:
            evenodd();
            break;
        default:
            wrong(menu_item);
            break;
    }
} while (menu_item != 5);
```

Κατά την κλήση των συναρτήσεων περνάμε κάποιες παραμέτρους έτσι ώστε όταν υλοποιηθούν οι εντολές της κάθε συνάρτησης να έχει τα δεδομένα για να κάνει τις σωστές πράξεις. Οι παράμετροι έχουν αποθηκευτεί σε μορφή `structure` κατά την εκκίνηση του προγράμματος μας, αντίστοιχα με μία συνάρτηση.

```
int main() {
    struct Input input = user();
    int menu_item = 0;
    do
    {
        .
        .
        .
        .
    }
    return 0;
}
```

```

struct Input user(){
    struct Input input;
    printf("Enter the first Integer number:");
    scanf("%d",&input.number_1);
    printf("Enter the second Integer number:");
    scanf("%d",&input.number_2);
    return input;
}

```

Ο λόγος που πάρθηκε η απόφαση να χρησιμοποιηθεί structure για την αποθήκευση των δύο ακέραιων μεταβλητών είναι διότι μία συνάρτηση μπορεί να επιστρέψει μόνο μία τιμή κάθε φορά. Μέσα στη συνάρτηση αποθηκεύουμε σε ένα structure τους δύο ακέραιους και στο return της συνάρτησης επιστρέφουμε όλο το structure.

Για κάθε λειτουργία του προγράμματός μας έχουμε δημιουργήσει μία συνάρτηση. Παρακάτω θα δούμε αναφορικά την κάθε συνάρτηση και την λειτουργία της.

Η συνάρτηση τύπου struct, Input user(),

```

struct Input user(){
    struct Input input;
    printf("Enter the first Integer number:");
    scanf("%d",&input.number_1);
    printf("Enter the second Integer number:");
    scanf("%d",&input.number_2);
    return input;
}

```

8

αρχικά δημιουργεί μία δομή structure και στη συνέχεια “ζητάει” από τον χρήστη να εισάγει δύο ακέραιους αριθμούς και τους αποθηκεύει στη δομή structure.

Η συνάρτηση τύπου int, menu(),

```

int menu(){
    int choice = 0;
    printf("\n");
    printf("Make your selection please\n");
    printf("=====\\n");
    printf("|| 1. Addition +           ||\\n");
    printf("|| 2. Subtraction -         ||\\n");
    printf("|| 3. Multiplication *       ||\\n");
    printf("|| 4. Even or Odd           ||\\n");
    printf("|| 5. Exit                   ||\\n");
    printf("=====\\n");
    scanf("%d",&choice);
    return choice;
}

```

εμφανίζει το μενού στον χρήστη και αποθηκεύει σε μία μεταβλητή την επιλογή του χρήστη. Τέλος, επιστρέφει σαν τιμή την επιλογή του χρήστη για να χρησιμοποιηθεί από το κυρίως πρόγραμμα.

Οι συναρτήσεις τύπου void, add(), sub(), mul(),

```
void add(int num_1,int num_2){
    int sum= num_1+num_2;
    printf("\n");
    printf(" ~ Addition Mode ~\n");
    printf("The Addition of %d and %d is: %d\n",num_1,num_2,sum);
    printf("\n");
}

void sub(int num_1,int num_2){
    int sub= num_1-num_2;
    printf("\n");
    printf(" ~ Substraction Mode ~\n");
    printf("The Substraction of %d and %d is: %d\n",num_1,num_2,sub);
    printf("\n");
}

void mul(int num_1,int num_2){
    int mul= num_1*num_2;
    printf("\n");
    printf(" ~ Multiplication Mode ~\n");
    printf("The Multiplication of %d and %d is: %d\n",num_1,num_2,mul);
    printf("\n");
}
```

παίρνουν σαν παραμέτρους τις δύο τιμές των ακέραιων αριθμών που είχε εισάγει ο χρήστης στην αρχή του προγράμματος και εκτελούν, πρόσθεση, αφαίρεση και πολλαπλασιασμό αντίστοιχα.

9

Η συνάρτηση τύπου void, evenodd(),

```
void evenodd(){
    int a = 0;
    printf("\n");
    printf(" ~ Even or Odd Mode ~\n");
    printf("Enter an integer number\n");
    scanf("%d",&a);
    if (a%2 == 0) //modulus operator
        printf("The given number %d is even\n",a);
    else
        printf("The given number %d is odd\n",a);
    printf("\n");
}
```

αποθηκεύει σε μία τοπική μεταβλητή έναν ακέραιο αριθμό που εισάγει ο χρήστης και με το modulus % operator ελέγχει αν ο αριθμός είναι άρτιος ή περιττός.

Τέλος, η συνάρτηση τύπου int, wrong(),

```
int wrong(int number){
    printf("\n");
    if (number != 5)
        printf("Wrong selection number\n");
}
```

ελέγχει τον αριθμό επιλογής του χρήστη και εμφανίζει το ανάλογο μήνυμα.

Γιαννουλάτος Λάμπρος

Α' Εξάμηνο
2022-2023