

11.1 Άσκηση 1

11.1.1 Περιγραφή

Να δημιουργηθεί πρόγραμμα σύμφωνα με το οποίο ο υπολογιστής να δέχεται δύο ακέραιους αριθμούς x και y και στη συνέχεια:

1. να υπολογίζει το άθροισμά τους,
2. να βρίσκει τον μεγαλύτερο,
3. να βρίσκει το x^y ,
4. να υπολογίζει τους δεκαδικούς λογαρίθμους των x και y .

Όλα τα παραπάνω να τα εμφανίζει στην οθόνη με κατάλληλα σχόλια.

Χρόνος εκτέλεσης: Κατά τη διάρκεια της μελέτης του κεφαλαίου 3 ή με την ολοκλήρωσή του.

11.1.2 Στόχος

- α) Εξοικείωση με το C μεταγλωττιστή και το σύστημα.
- β) Εξοικείωση με δήλωση μεταβλητών, εισαγωγή δεδομένων από την κύρια είσοδο, εκτέλεση βασικών διεργασιών.

11.1.3 Οδηγίες

Μελετώντας το πρόβλημα εντοπίζουμε τις βασικές διεργασίες που το σύστημα πρέπει να εκτελεί. Οι διεργασίες αυτές είναι:

- εισαγωγή αριθμού
- αποθήκευση αριθμού
- υπολογισμός αθροίσματος
- εύρεση μεγαλύτερου
- υπολογισμός δύναμης
- υπολογισμός λογαρίθμου

Εισαγωγή αριθμού από την κύρια είσοδο

Για την εισαγωγή αριθμού, και γενικότερα για είσοδο και έξοδο δεδομένων, η C δε διαθέτει άμεση υποστήριξη με τη μορφή εντολών ή τελεστών. Οι λειτουργίες εισόδου/εξόδου υποστηρίζονται από συναρτήσεις της βασικής βιβλιοθήκης. Μεταξύ των συναρτήσεων αυτών διακρίνουμε τις `getchar` και `scanf` για είσοδο και τις `putchar` και `printf` για έξοδο.

11.1.3.1 Συν

Η συνάρτηση
στο πρόγραμ
από τη δεσμη

Για τη χρή
`stdio.h`. Αυτά
την παρακάτ

```
#include
```

Γιατί να συμ

Ο τρόπος χρή
αριθμός και
επιστρεφόμε
prototype). Τ
(interface) τη
γος που συμπ
σιμοποιήσω τ

Το δεύτερο
implementati
βασικής βιβλ
ριλάβει στο π

Το πρότυπο

```
int getcha
```

Αυτό σημαίν
όρισμα (δηλά
ακέραια.

Την επιστρ
σης σε μία μ
δηλώνει τη μ
που η συνάρτ

```
char ch;
```

```
:
```

```
ch = getch
```

11.1.3.1 Συνάρτηση getch

Η συνάρτηση διαβάζει ένα χαρακτήρα από την κύρια είσοδο και τον επιστρέφει στο πρόγραμμα. Ώς κύρια είσοδος θεωρείται το πληκτρολόγιο και παρίσταται από τη δεξιμήλη λέξη `stdin`.

Για τη χρήση της είναι απαραίτητη η συμπεριληψη του αρχείου πηγαίου κώδικα `stdio.h`. Αυτό γίνεται περιλαμβάνοντας στην αρχή του αρχείου πηγαίου κώδικα την παρακάτω εντολή του προ-επεξεργαστή:

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

Γιατί να συμπεριλάβω το `stdio.h`;

Ο τύπος `char` της συνάρτησης, δηλαδή το όνομα με το οποίο την καλούμε, ο αριθμός και ο τύπος των ορισμάτων που τις περνάμε καθώς και ο τύπος της επιστροφής της, ορίζεται από το πρωτότυπο της συνάρτησης (function prototype). Το πρωτότυπο της συνάρτησης που αποτελεί στην πράξη τη διεπαφή (interface) της συνάρτησης βρίσκεται στο αρχείο `stdio.h`. Αυτός είναι και ο λόγος που συμπεριλαμβάνω το αρχείο αυτό στον πηγαίο κώδικα όταν θέλω να χρησιμοποιήσω τη συνάρτηση.

Το δέτερο τιμήα της συνάρτησης που είναι η υλοποίηση της (function implementation) βρίσκεται υπό μορφή λίστας `lib` αρχείο της βασικής βιβλιοθήκης. Είναι εθνη του linker να το εντοπιστεί και να το συμπεριλάβει στο παραγόμενο εκτελέσιμο πρόγραμμα.

Το πρότυπο της συνάρτησης `getchar` είναι:

```
int getch(void);
```

Αυτό σημαίνει ότι κάλώ τη συνάρτηση με όνομα `getchar`, δεν της περνά κανένα όρισμα (δηλώνεται από τη λέξη κλειδί `void`) και η επιστροφή της είναι ακέραια.

Την επιστροφή της τη μπορώ να την καταχωρήσω με μία πρόταση ανάθεσης σε μία μεταβλητή int ή `char` για να παραπαραχρηση. Ο παρακάτω κώδικας δηλώνει τη μεταβλητή `ch`, στη συνέχεια καλεί τη `getchar` και αναθέτει την τιμή που η συνάρτηση θα πάρει από την κύρια είσοδο στη μεταβλητή `ch`.

```
char ch;
/* δήλωση μεταβλητής ch */
ch = getch();
/* κλίση της συνάρτησης getch και ανάθεση της
/* επιστροφής της στη μεταβλητή ch */
```

δομένων, η C
 και λειτουργίες
 βιβλιοθήκης.
 για είσοδο

το σύνολο

την κύρια εί-

η τιμή του

να δέχεται

11.1.3.2 Συνάρτηση scanf

Η συνάρτηση διαβάζει από την κύρια είσοδο δεδομένα διαφόρων τύπων και τα καταχωρεί σε αντιστοίχου τύπου μεταβλητές. Για τη χρήση της απαιτείται συμπερίληψη του αρχείου `stdio.h`

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

Το πρωτότυπο της είναι¹:

```
int scanf(const char *format, όρισμα2, όρισμα3, όρισμα4, ...);
```

Το πρώτο όρισμα ορίζει σύμφωνα με δεδομένους κανόνες (βλέπε εγχειρίδιο βασικής βιβλιοθήκης) τον αριθμό και τον τύπο των δεδομένων που θα διαβάσει η συνάρτηση. Τα ορίσματα που ακολουθούν είναι διευθύνσεις μεταβλητών στις οποίες η συνάρτηση θα αποθηκεύσει την πληροφορία που διαβάζει. Στη συνέχεια, δίνονται ορισμένα παραδείγματα χρήσης της.

A) Ανάγνωση χαρακτήρα

```
char ch;
scanf("%c",&ch);
```

Το σύμβολο `&` είναι ο τελεστής άμεσης διεύθυνσης ο οποίος, όταν εφαρμόζεται σε μία μεταβλητή, δίνει τη διεύθυνση της μεταβλητής.

B) Ανάγνωση χαρακτήρα και ακεραίου

```
char ch;
int num;
:
scanf("%c %d",&ch,&num);
```

Η `scanf` διαβάζει ένα χαρακτήρα (`%c`) και τον αποθηκεύει στη `ch` και ένα δεκαδικό (`%d`) και τον αποθηκεύει στη `num`.

Γ) Ανάγνωση αλφαριθμητικού

```
char name[30];          /* δηλώνει ένα πίνακα χαρακτήρων */
:
scanf("%s",name);
```

Προσέξτε ότι δε χρησιμοποιούμε τον τελεστή `&`. Αυτό συμβαίνει γιατί το όνομα του πίνακα αντιπροσωπεύει τη διεύθυνση του πρώτου στοιχείου του πίνακα.

¹ Για να είμαστε πιο ακριβείς, το πρωτότυπο είναι `int scanf(char *, ...);`

11.1.3.3 Συν

```
#include
```

Το πρωτότυ

```
int putchar
```

Η συνάρτηση
έξοδο και ε
τιμή EOF σε
στο αρχείο s

```
#define
```

11.1.3.4 Συν

```
#include
```

Το πρωτότυ

```
int printf
```

Το πρώτο όρ
σικής βιβλιο
συνάρτηση
λοιπα ορίσμ
στην έξοδο.

A) Έξοδος α

```
printf("Δ
```

B) Έξοδος χ

```
char ch=  
printf("%
```

11.1.3.3 Ζωγράφηση putchar

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

Το πρώτο τμήρο της είναι:

```
int putchar(int ch);
```

Η συνάρτηση εξάγει το χαρακτήρα που περιέχει η μεταβλητή `ch` στην κβία `έξοδο` και επιστρέφει το χαρακτήρα που έγραψε για επτυχή λειτουργία ή την τιμή EOF σε περίπτωση λάθους. Η δεσμευμένη λέξη EOF (End Of File) ορίζεται στο αρχείο `stdio.h` ως

```
#define EOF -1
```

11.1.3.4 Ζωγράφηση printf

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

Το πρώτο τμήρο της είναι:

```
int printf(const char *format, όπισμα2, όπισμα3, όπισμα4, ...);
```

Το πρώτο όπισμα ορίζει σήμφωνα με δεδομένους κανόνες (βλέπετε εγχειρίδιο β-σικής βιβλιοθήκης) τον αριθμό και τον τύπο των όπισμάτων που θα εξάγει η συνάρτηση στην κβία `έξοδο`. Αυτά θα πρέπει να είναι σε σήμφωνα με τα υπόλοιπα όπισματα που ακολούθούν και τις τιμές των όπιστων η συνάρτηση βγάξει στην `έξοδο`. Στη συνέχεια, δίνονται όπισμένα παραδειγματα χρήσης της:

A) Έξοδος αφαιρεθιμητικής σταθεράς

```
printf("Δώσε ένα χαρακτήρα:");
```

B) Έξοδος χαρακτήρα

```
char ch='A';  
printf("%c\n", ch); /* η πόδιαση τυπώνει ως χαρακτήρα (%c) */  
/* το περιέχόμενο του ch και στη συνέχεια */  
/* αλλαζει γραμμή(\n) */
```

```
printf("%c \t %c", ch, 'B');
```

Η τελευταία πρόταση τυπώνει ως χαρακτήρα (%c) το περιεχόμενο του ch, στη συνέχεια τυπώνει ένα tab (\t) και τέλος τυπώνει ως χαρακτήρα (%c) το χαρακτήρα 'B'.

Γ) Έξοδος χαρακτήρα και ακεραίου

```
char ch = 'a';
int num = 1821;
printf("Ο χαρακτήρας είναι: %c , ο ακέραιος είναι: %d \n",ch,num);
```

Η πρόταση τυπώνει ως χαρακτήρα (%c) το περιεχόμενο του ch, ως δεκαδικό (%d) το περιεχόμενο της num και στη συνέχεια αλλάζει γραμμή (\n).

Δ) Έξοδος χαρακτήρα, ακεραίου και αλφαριθμητικού

```
char ch = 'a';
int num = 1821;
char name[20] = "abcd";
printf("Ο χαρακτήρας είναι: %c , ο ακέραιος είναι: %d, το αλφαριθμητικό είναι: %s \n",ch,num,name);
```

Η πρόταση τυπώνει ως χαρακτήρα (%c) το περιεχόμενο του ch, ως δεκαδικό (%d) το περιεχόμενο της num, ως αλφαριθμητικό τη name και στη συνέχεια αλλάζει γραμμή (\n).

11.1.4 Αποθήκευση αριθμού

Για την αποθήκευση του εισαγόμενου με τη βοήθεια της scanf αριθμού, θα πρέπει να έχει δηλωθεί η κατάλληλη μεταβλητή. Τη λειτουργία της αποθήκευσης την εκτελεί η ίδια η scanf. Για παράδειγμα, για εισαγωγή ακεραίου αριθμού, θα πρέπει να δηλώσω μια μεταβλητή int με την πρόταση δήλωσης

```
int num;
```

και στη συνέχεια να καλέσω τη scanf για να διαβάσει τον αριθμό από την κύρια είσοδο και να τον τοποθετήσει στην num

```
scanf("%d, &num);
```

11.1.5 Υπο

Η διεργασία
τον τελεστή
ρώ να το ερμ

```
printf("
```

ή να το ανα

```
sum = r
```

11.1.6 Εύρ

Για την εύρ
έναν από τα
Με τη χ

```
num1 >
```

η οποία είν
είναι ψευδ

Στη συν
πρόταση ελ
γίες, π.χ. ν

```
if(num1
print
else
print
```

ή να αναθέ

```
if(num1
max=
else
max=
```

και στη συν

11.1.7 Υπολογισμός δύναμης και λογαρίθμου

Οι διεργασίες δεν υποστηρίζονται άμεσα από τη γλώσσα C. Υποστηρίζονται όμως από τις συναρτήσεις της βασικής βιβλιοθήκης `pow` και `log10`.

11.1.7.1 Συνάρτηση `pow`

Για να χρησιμοποιηθεί η συνάρτηση `pow`, πρέπει να συμπεριληφθεί το αρχείο επικεφαλίδας `math.h`. Αυτό γίνεται περιλαμβάνοντας στην αρχή του πηγαίου κώδικα αμέσως μετά τα σχόλια την πρόταση:

```
#include <math.h>
```

Το πρωτότυπο της συνάρτησης είναι:

```
double pow(double x, double y);
```

Η συνάρτηση υπολογίζει το x στην y . Σχόλια και παράδειγμα χρήσης της συνάρτησης θα βρείτε μετά την παρουσίαση της συνάρτησης `log10`.

11.1.7.2 Συνάρτηση `log10`

Για να χρησιμοποιηθεί η συνάρτηση `log10`, πρέπει να συμπεριληφθεί το αρχείο `math.h`. Αυτό γίνεται περιλαμβάνοντας στην αρχή του πηγαίου κώδικα αμέσως μετά τα σχόλια την πρόταση:

```
#include <math.h>
```

Το πρωτότυπο της συνάρτησης είναι:

```
double log10(double x);
```

Η συνάρτηση υπολογίζει το δεκαδικό λογάριθμο του x .

Για το σύστημα UNIX κατά την διαδικασία μεταγλώττισης χρησιμοποιείτε το `option -lm` του μεταγλωττιστή.

Στη συνέχεια, δίνεται ο πηγαίος κώδικας ενός προγράμματος υπολογισμού αθροίσματος δύο ακεραίων.

➤
Οι συ
τους
σματ
ναγκ
σμό
H

pow

και
μετα

doubl
:
pow
:
print

ή εν

print
(dou

11.1.8 Ενδ

```
/* *****  
/* addin  
/* Πρόγ  
/* *****  
  
#includ  
  
main()  
{  
int num  
int resu
```

➤ **Συναρτήσεις pow, log10**

Οι συναρτήσεις pow και log10 δέχονται ορίσματα τύπου double. Αν τους περάσουμε ορίσματα τύπου int, έχουμε λάθος, να αποφευχθεί. Για την ορθή κλήση των συναρτήσεων, θα πρέπει να εξασφαλίσουμε σε μεταβλητή τύπων χρησιμοποιώντας το ηχητικό casting.

Η κλήση της pow θα διαμορφωθεί όπως κατωτέρω:

```
pow((double)x,(double)y);
```

και σποδήποτε το αποτέλεσμα θα πρέπει να εκχωρηθεί σε μεταβλητή τύπου double για να μην έχουμε λάθος.

```
double power;
```

```
power = pow((double)x,(double)y);
```

```
printf("Το %d στην %d είναι %f\n", x,y,power);
```

ή αναλλακτικά αποφεύγοντας τη χρήση μεταβλητής

```
printf("Το %d στην %d είναι %f\n", x,y,pow((double)num1, (double)num2));
```

11.1.8 Ενδεικτικός πηγαίος κώδικας

```

/* *****
 * addint.c
 *
 * Πρόγραμμα υπολογισμού αθροίσματος δύο ακέραιων
 *
 * *****
#include <stdio.h>
main()
{
int num1,num2;
int result;

```

λογισμίου
 ορισίστε
 α αριθμός
 το αρχείο
 5 συνάρ-
 πηγαίου
 το αρχείο
 πταθμός