

## 11.3 Άσκηση 3-Τέσσερις πράξεις

### 11.3.1 Περιγραφή

Η άσκηση δίνεται σε δύο εκδόσεις. Η δεύτερη, η 3β, προϋποθέτει την εκτέλεση και πλήρη κατανόηση της πρώτης έκδοσης 3α.

### 11.3.2 Άσκηση 3α

#### Βήμα 1

Να δημιουργηθεί πρόγραμμα το οποίο να δίνει τη δυνατότητα στο χρήστη να εισάγει δύο αριθμούς και στη συνέχεια να εκτελεί πάνω σε αυτούς διάφορες πράξεις (π.χ. πρόσθεση, αφαίρεση, πολλαπλασιασμό, υπολογισμό δεκαδικού λογαρίθμου, κ.λ.π.) και να τυπώνει το αποτέλεσμα.

#### Βήμα 2

Τροποποιήστε το παραπάνω πρόγραμμα ώστε να επαναλαμβάνεται έως ότου ο χρήστης αποφασίσει τον τερματισμό της επανάληψης.

#### Βήμα 3

Προσθέστε έλεγχο της δια του μηδενός διαίρεσης και εμφανίστε ανάλογο μήνυμα.

#### Βήμα 4

Εντοπίστε βασικές διεργασίες και αναπαραστήστε αυτές από κατάλληλες συναρτήσεις. Διαμορφώστε το πρόγραμμα έτσι ώστε να αποτελείται από 3 τουλάχιστο συναρτήσεις.

#### Βήμα 5

Βελτιώστε την αναγνωσιμότητα του πηγαίου κώδικα της άσκησης χρησιμοποιώντας αλφαριθμητικά της μορφής (ADD, SUB, MUL και DIV) για την αναπαράσταση των επιλογών του χρήστη. Δώστε δύο εκδόσεις, μια με χρήση της εντολής #define του προ-επεξεργαστή και μια δεύτερη με κατάλληλη χρήση του αριθμητικού τύπου της C.

### 11.3.3 Άσκηση 3β

Η άσκηση 3β, η οποία προϋποθέτει την πλήρη κατανόηση της άσκησης 3α, είναι όπως η άσκηση 3α με τις παρακάτω διαφορές:

#### Βήμα 1

Να δημιουργηθεί πρόγραμμα που χρησιμοποιεί τις συναρτήσεις που υποστηρίζει το σύστημα.

Πιο αναλυτικά, να χρησιμοποιηθούν οι συναρτήσεις `system()`, `getchar()`, `printf()`, `scanf()`, `atoi()`, `atof()`, `DATEIME`.

#### Βήμα 2

Όπως το βήμα 1.

#### Βήμα 3

Όπως το βήμα 1.

#### Περιορισμοί

Για την άσκηση 3α.

```
int TCPdata;
int TCPerror;
```

Όπου

*host* είναι ο υπολογιστής που προσφέρει το *service* το οποίο ζητείται.

Το πρωτότυπο έχει ως εξής:

```
/* file: ex1.c
int TCPdata;
int TCPerror;
```

Το πρόσθετο

A) το αρχείο `ex1.c`  
B) τη βιβλιοθήκη `ex1.o` και `TCPecho.o`,

- κατεβάστε το αρχείο `ex1.c`
- στο σύνολο των αρχείων

☞ Σχόλιο.  
την Μάννα

### Βήμα 1

Να δημιουργηθεί πρόγραμμα το οποίο θα δίνει τη δυνατότητα στο χρήστη να χρησιμοποιεί τις υπηρεσίες ECHO και DATETIME που παρέχει ένα σύστημα που υποστηρίζει το πρωτόκολλο TCP.

Πιο αναλυτικά, να μπορεί να διαλέγει ένα σύστημα της απεσκέιας του π.χ. amalia και στη συνέχεια να επιδεικνύει τη χρήση των υπηρεσιών ECHO και DATETIME.

### Βήμα 2

Όπως το βήμα 2 της άσκησης 3α

### Βήμα 3

Όπως το βήμα 5 της άσκησης 3α.

### Περαιτέρω σχέδια

Για την άσκηση 3β, χρησιμοποιείστε τις παρακάτω συναρτήσεις:

```
int TCPdaytime(const char *host, const char *service);  
int TCPEcho(const char *host, const char *service);
```

Όπου

host είναι το όνομα της μηχανής από την οποία ζητάτε την υπηρεσία και service το όνομα της υπηρεσίας δηλαδή "echo", "daytime".

Το πρώτο των συναρτήσεων αυτών βρίσκεται στο αρχείο `ex2btcp.h` το οποίο έχει ως εξής:

```
/* file: ex2btcp.h */  
int TCPdaytime(const char *host, const char *service);  
int TCPEcho(const char *host, const char *service);
```

Το πρόθετο υλικό που απαιτείται για την άσκηση 3β, δηλαδή

- A) το αρχείο `ex2bcp.h` και
- B) τη βιβλιοθήκη `ex2bcp.a`, η οποία περιέχει τις συναρτήσεις `TCPdaytime` και `TCPEcho`, θα βρείτε
- κατεβάζοντας το αρχείο `exercis2b.zip` από το δικτυακό τόπο του βιβλίου ή
- στο συνοδευτικό CD.

### 11.3.4 Στόχος

Εξοικείωση με

- τις προτάσεις ελέγχου της ροής εκτέλεσης του προγράμματος
- την έννοια του μενού
- την έννοια του ελεγχόμενου τερματισμού προγράμματος
- δήλωση και ορισμό συναρτήσεων

**Χρόνος εκτέλεσης:** Κατά τη διάρκεια της μελέτης του κεφαλαίου 6 ή με την ολοκλήρωσή του.

### 11.3.5 Τρόπος αντιμετώπισης άσκησης 3α

#### Βήμα 1

Στο πρώτο αυτό βήμα θα αναπτύξετε ένα πρόγραμμα σύμφωνα με το οποίο το σύστημα θα εμφανίζει τις πράξεις/λειτουργίες (απόδοσης τιμής, πρόσθεσης, αφαίρεσης, διαίρεσης, κ.λ.π) που έχει τη δυνατότητα να εκτελέσει πάνω στους δύο αριθμούς και θα δίνει τη δυνατότητα στο χρήστη να επιλέξει.

Ο χρήστης θα επιλέγει, θα γνωρίζει την επιλογή του στο σύστημα και έτσι αυτό θα εκτελεί την αιτούμενη πράξη και θα εμφανίζει το αποτέλεσμα.

#### Οδηγίες

Εμφανίστε στην οθόνη με κατάλληλη χρήση της printf, τις πράξεις που το πρόγραμμα εκτελεί. Αναθέστε σε κάθε πράξη ένα χαρακτήρα (π.χ. A for Add) ή ένα αριθμό (π.χ. 1 - Add). Ενημερώστε το χρήστη ότι μπορεί να επιλέξει μία από τις λειτουργίες αυτές πατώντας τον αντίστοιχο αριθμό/χαρακτήρα. Αυτό μπορεί να γίνει με πρόταση της μορφής

```
printf("Επέλεξε μία από τις λειτουργίες 1-5:");
```

Διαβάστε την επιλογή του με κατάλληλη χρήση της scanf ή getch. Δηλώστε μια μεταβλητή *choice* τύπου *char* ή *int* για να αποθηκεύσετε την επιλογή του χρήστη.

Χρησιμοποιήστε την πρόταση *switch()* για να αποφασίσετε ποια λειτουργία θα εκτελέσετε. Εκτελέστε τη λειτουργία και εμφανίστε το αποτέλεσμα.

Ανατρέξτε στην ενότητα 6.2 για να δείτε πώς χρησιμοποιείται η πρόταση *switch*.

### Βήμα 2

Για να αποφευχθεί η πράξη, το σύστημα θα λειτουργεί του χρήστη απ

#### Οδηγίες

Χρησιμοποιήστε τον *switch* τρέψετε τον *switch* δείτε πώς χ

☞ Πρ  
πισμό τ  
ενσωμα  
επανάλ  
πρότασ

#### Πρόταση if

Η σύνταξη τ

```
if ( έ  
πρότ
```

ή εναλλακτικ

```
if ( έ  
πρότ  
else  
πρότ
```

Στο παρακάτ

```
if (count  
:  
}  
< επόμεν
```

## Βήμα 2

Για να αποφύγουμε την εξάσφαλη εκτέλεση του προγράμματος για εκτέλεση νέας πράξης, το πρόγραμμα πρέπει να τροποποιηθεί ώστε να μην τερματίζει τη λειτουργία του μετά την εμφάνιση του πρώτου αποτρεξήματος αλλά μόνο όταν ο χρήστης αποφασίσει για τον τερματισμό του.

## Οδηγίες

Χρησιμοποιήστε μια από τις ποσότητες επανάληψης (*for*, *while*, *do*) για να αποφύγετε τον τερματισμό του προγράμματος. Αναρξέξτε στην ενότητα 6.2 για να δείτε πώς χρησιμοποιούνται οι ποσότητες *for*, *while* και *do*.

**Προσοχή.** Θα πρέπει να αποφάξτε τρόπο για κανονικό τερματισμό. Θα πρέπει να αποφάξτε τη χρήση της ενότητας τερματισμού επανάληψης ή χρησιμοποίηση της ενότητας *if* σε συνδυασμό με την ενότητα *break*.

## Πρόταση if

Η συνταξη της είναι

```
if ( εκφραση )  
    πρόταση
```

ή εναλλακτικά

```
if ( εκφραση )  
    πρόταση
```

else

πρόταση

Στο παρακάτω τμήμα κώδικα

```
if (count < MAX_COUNT) {  
    :  
    }  
< επόμενη πρόταση >
```

το σύστημα υπολογίζει την τιμή της έκφρασης

```
count < MAX_COUNT
```

και, αν η τιμή αυτή είναι αληθής, εκτελεί το σώμα της `if`, διαφορετικά ο έλεγχος μεταφέρεται στην

<επόμενη πρόταση>

Στο παρακάτω τμήμα κώδικα

```
if (num[i]<0) {
    printf("Ο αριθμός είναι αρνητικός");
    :
}
else {
    printf("Ο αριθμός είναι θετικός");
    :
}
```

το σύστημα υπολογίζει την τιμή της έκφρασης

```
num[i]<0
```

και, αν η τιμή αυτή είναι αληθής, το σύστημα εκτελεί την

```
printf("Ο αριθμός είναι αρνητικός");
```

διαφορετικά εκτελεί την

```
printf("Ο αριθμός είναι θετικός");
```

Σε κάθε περίπτωση, μετά ο έλεγχος μεταφέρεται στην

<επόμενη πρόταση>

### Πρόταση *break*

Η **break** προκαλεί την έξοδο από το βρόχο επανάληψης που την περικλείει. Θα πρέπει να σημειωθεί ότι προκαλεί την έξοδο μόνο από το πρώτο βρόχο που την περικλείει.

Στο παρακάτω

```
for( i = 0
    if (nu
    :
    }
    printf(...
```

το σύστημα  
Αν ναι, εκτε  
να μεταφερε

### Βήμα 4

Για τη δήλωση  
τα του πηγα  
Διαβάζο  
γειες κάνει  
ραίτη η χ

### Βήμα 5

Μελετήστε  
τύπο της C.  
τερη με κατ  
θεια, μπορε

### 11.3.6 Ενδ

```
/* *****
/* test3.
/* Πρόγρ
/* *****
#include
#include
```

```
float nu
double n
```

```
void ma
{
```

Στο παρακάτω τμήμα κώδικα

```

for ( i = 0 ; i < MAX ; i++) {
    if ( num[i] < 0 ) /* end loop */
        break;
    /* process positive elements */
    printf(.....);
}

```

το σβήσιμο ελέγχει σε κάθε επανάληψη, αν το στοιχείο *num[i]* είναι αρνητικό. Αν ναι, εκτελεί την *break* με αποτέλεσμα να τερματιστεί ο βρόχος και ο έλεγχος να μεταφερθεί στην *printf*.

#### Βήμα 4

Για τη δήλωση των συναρτήσεων, δώστε ιδιαίτερη έμφαση στην αναγνώσιμότητά του προγράμματος κώδικα. Διαβάζοντας τον κώδικα της *main* θα πρέπει να καταλάβετε τι ενέργειες κάνει το σβήσιμο όταν εκτελεί το πρόγραμμα και αυτό χωρίς να είναι απαραίτητη η χρήση σχολίων.

#### Βήμα 5

Μετατήστε την πρόταση *#define* του προ-επεξεργαστή και τον αριθμητικό τύπο της *C*. Δώστε δύο εκδόσεις, μια με χρήση της εντολής *#define* και μια δέυτερη με κατάλληλη χρήση του αριθμητικού τύπου της *C*. Αν χρειαστείτε βοήθεια, μπορείτε να ανατρέξετε στο παράδειγμα 6-7 του βιβλίου.

### 11.3.6 Ενδεικτικός πηγαίος κώδικας

```

/* *****
 * test3.c - calculator
 *
 * Πρόγραμμα εκτέλεσης διαφόρων πράξεων πάνω σε δύο αριθμούς
 *
 * *****
#include <stdio.h>
#include <math.h>
float num1,num2;
double result;
void main()
{

```

```

char choice;

while(...){

    /* Εμφάνιση του μενού επιλογής */
    :
    :
    /* Δέχεται την επιλογή του χρήστη */
    :
    /* Ελέγχει για τερματισμό του προγράμματος */
    if(...)
        break;

    /* Εκτέλεση λειτουργίας */
    switch(choice){
        case 1 :           /* Εισαγωγή των δύο αριθμών */
            :
            :
            break;
        case 'A' : /* Add */
            :
            :
            break;
        <υπόλοιπα case>
            :
            :

        default :
            printf("Unknown command\n");

        break;
    }
}

```

## 11.4 Ασκήσεις

### 11.4.1 Παράδειγμα

Να γραφεί πρόγραμμα που θα διαβάζει τις παραμέτρους

1. Εισαγωγή των δύο αριθμών
2. Εμφάνιση του μενού επιλογής
3. Ταξινόμηση των αριθμών
4. Ταξινόμηση των αριθμών
5. Εμφάνιση του μενού επιλογής

- μέγεθος
- ελάχιστο
- μέγιστο

6. Εισαγωγή των δύο αριθμών

- σύμμετρο
- μέγεθος
- μέγιστο
- τη
- τη

7. Εισαγωγή των δύο αριθμών

εμφάνιση του μενού επιλογής

### 11.4.2 Στοιχεία

Εξοικειώστε

- 
- 
- 
- 

*Χρόνος εκτέλεσης*

*κλήρωσή του*