

Α' Εξάμηνο

ΕΙΣΑΓΩΓΗ ΣΤΟ ΔΟΜΗΜΕΝΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ

Εργαστήριο – 3η σειρά ασκήσεων.

Κοζάνη, 26 Οκτωβρίου 2007.

Πρόγραμμα p3-1 (υλοποίηση ψευδοκώδικα σε πρόγραμμα)

Εύρεση του μέγιστου, ελάχιστου και μέσου όρου μιας σειράς θετικών αριθμών που δίνει ο χρήστης.

Όρισε και θέσε $sum = min = max = n = 0$

Ζήτησε θετικό αριθμό και μηδέν ή αρνητικό για να τερματίσεις το πρόγραμμα

Διάβασε x

Θέσε $min = x$

Θέσε $n = 1$

Αν x είναι θετικό κάνε:

Αν $x < min$ θέσε $min = x$

Αν $x > max$ θέσε $max = x$

Αύξησε το sum κατά x

Αύξησε το n κατά 1

Ζήτησε άλλον έναν αριθμό

Διάβασέ τον

Συνέχισε

Τύπωσε ελάχιστο, μέγιστο και μέσο όρο (sum / n)

Πρόγραμμα p3-2 (υλοποίηση ψευδοκώδικα)

Το ίδιο όπως πριν, με προκαθορισμένο αριθμό επαναλήψεων. Δεν υπάρχει περιορισμός ως προς το πρόσημο.

Όρισε sum, min, max

Ζήτησε αριθμό πλήθους όρων

Διάβασε n

Ζήτησε αριθμό

Διάβασε x

Θέσε $min = max = sum = x$

Κάνε $n-1$ φορές:

Ζήτησε αριθμό

Διάβασε x

Αν $x < min$ θέσε $min = x$

Αν $x > max$ θέσε $max = x$

Αύξησε το sum κατά x

Τύπωσε ελάχιστο, μέγιστο και μέσο όρο (sum / n)

Πρόγραμμα p3-3 (άσκηση εκτέλεσης)

Παραλλαγή των προηγούμενων δύο, αλλά εδώ εισάγεται και η έννοια του πίνακα (array). Το πρόγραμμα αποθηκεύει τα δεδομένα σε ένα μονοδιάστατο πίνακα και μετά τα επεξεργάζεται. Δεν υπάρχει περιορισμός ως προς το πρόσημο.

Μεταγλωττίστε και εκτελέστε:

```
#include <stdio.h>
#define NDAT 10

int main()
{
    int n;
    float data[NDAT];
    float sum, lo, hi;

    for(n = 0; n < NDAT; n++)
        data[n] = 0.0;

    for(n = 0; n < NDAT; n++)
    {
        printf("Δώσε μου το υπ' αριθ. %d στοιχείο: ", n+1);
        scanf("%f", &data[n]);
    }

    lo = hi = data[0];
    for(n = 0; n < NDAT; n++)
    {
        sum += data[n];
        lo = (lo > data[n]) ? data[n] : lo;
        hi = (hi < data[n]) ? data[n] : hi;
    }
    printf("\nΚυμαίνονται μεταξύ %f και %f και έχουν μέσο όρο %f\n", lo, hi, sum/n);
}
```

Πρόγραμμα p3-4 (επίλυση προβλήματος)

Διαβάστε τους στίχους από το τραγούδι “99 bottles on the wall”. Η ομοιότητα κάθε στροφής με τις υπόλοιπες είναι τόσο μεγάλη ώστε κάποιος παρατήρησε ότι η εκτύπωση όλων των στίχων είναι άσκοπη αφού θα μπορούσε να γραφτεί ένα μικρό πρόγραμμα που να τους αναπαράγει αυτόματα! Από τότε πολλοί μπήκαν στον πειρασμό και έγραψαν τέτοια μικρά προγράμματα σε διάφορες γλώσσες. Μέχρι σήμερα έχουν γραφτεί προγράμματα σε 1127 διαφορετικές γλώσσες προγραμματισμού που αναπαράγουν τις στροφές αυτού του τραγουδιού. Όλα είναι δημοσιευμένα στον ιστότοπο <http://www.99-bottles-of-beer.net>.

Ζητείται να γράψετε ένα τέτοιο μικρό πρόγραμμα σε C χρησιμοποιώντας μία δομή επανάληψης

τύπου for.

Πρόγραμμα p3-5 (επίλυση προβλήματος)

Το ίδιο με το προηγούμενο χρησιμοποιώντας άλλη δομή επανάληψης αντί για for (while ή do while).

Σχόλιο: αυτή η διαδικασία μπορεί να θεωρηθεί και ως μία μορφή συμπίεσης πληροφορίας.

Βλέποντάς το από μια άλλη σκοπιά, μπορούμε επίσης να πούμε ότι το τραγούδι αυτό έχει χαμηλή *αλγοριθμική πολυπλοκότητα* (μέτρο της αλγοριθμικής πολυπλοκότητας μιας συμβολοσειράς είναι το μήκος του μικρότερου προγράμματος που αρκεί για να την αναπαράγει).

Πρόγραμμα p3-6 (βελτίωση προγράμματος)

Το παρακάτω πρόγραμμα είναι σωστό αλλά όχι και τόσο καλογραμμένο. Μπορείτε να το ξαναγράψετε καλύτερα;

```
#include <stdio.h>

int main( )
{
    float x;

    printf("\nΔώσε μου έναν πραγματικό αριθμό: ");
    scanf("%f", &x);

    if( x == 0 )
        printf(" x = %f δίνει παρονομαστή μηδέν!\n", x);
    else {
        if( x == 1 )
            printf(" x = %f δίνει παρονομαστή μηδέν!\n", x);
        else {
            if( x == 2 )
                printf(" x = %f δίνει παρονομαστή μηδέν!\n", x);
            else
                printf("Το αποτέλεσμα είναι %f \n", 1.0/(x*(x-1.0)*(x-2.0)));
        }
    }
}
```

ΑΠΑΝΤΗΣΕΙΣ

Πρόγραμμα p3-1

```
#include <stdio.h>

int main()
{
```

```

int n;
float sum, min, max, x;

sum = max = n = 0;
/* το min θα πάρει τιμή πιο κάτω` δε χρειάζεται εδώ */

printf("Δίνε μου θετικούς αριθμούς και 0 ή αρνητικό για
τερματισμό.");
printf("\nΔώσε μου τον πρώτο αριθμό ");
scanf("%f", &x);
min = x;
while (x > 0.0)
{
    if(x < min) min = x;
    if(x > max) max = x;
    sum += x;
    n++;
    printf("\nΔώσε άλλον έναν αριθμό: ");
    scanf("%f", &x);
}
printf("\nΟ ελάχιστος είναι %f και ο μέγιστος %f", min, max );
printf("\nΟ μέσος όρος είναι %f\n", sum/n);
}

```

Πρόγραμμα p3-2

Στα σχόλια δίνονται εναλλακτικές μορφές της εντολής επανάληψης που θα έδιναν το ίδιο αποτέλεσμα.

```

#include <stdio.h>

int main()
{
    int i, n;
    float sum, min, max, x;

    printf("Δώσε μου πλήθος όρων: ");
    scanf("%d", &n);
    printf("\nΔώσε μου τον όρο υπ' αριθ. 1: ");
    scanf("%f", &x);
    min = max = sum = x;
    /*
    for (i = 0; i < n-1; i++)
    for (i = 1; i < n; i++)

```

```

for (i = n; i > 1; i--)
for (i = n-1; i > 0; i--)  */
for (i = 2; i <= n; i++)
{
    printf("\nΔώσε μου τον όρο υπ' αριθ. %d: ", i);
    scanf("%f", &x);
    if(x < min) min = x;
    if(x > max) max = x;
    sum += x;
}
printf("\nΟ ελάχιστος είναι %f και ο μέγιστος %f", min, max );
printf("\nΟ μέσος όρος είναι %f\n", sum/n);
}

```

Προσοχή! Η διαφορά των δύο προγραμμάτων p3-1 και p3-2, όσον αφορά τα δεδομένα εισόδου, δε σημαίνει ότι η εντολή while υστερεί από αυτή την άποψη έναντι της for. Αντίθετα, μία δομή for μπορεί να “προσομοιωθεί” και από μία while, όπως αποδεικνύει και η παρακάτω εναλλακτική λύση (με έντονο κόκκινο οι εντολές που υποκαθιστούν τη δομή for του προηγούμενου προγράμματος). Γενικά, μια δομή while είναι πιο γενική από μια for.

Πρόγραμμα p3-2a

```

#include <stdio.h>

int main()
{
    int i, n;
    float sum, min, max, x;

    printf("Δώσε μου πλήθος όρων: ");
    scanf("%d", &n);
    printf("\nΔώσε μου τον όρο υπ' αριθ. 1: ");
    scanf("%f", &x);
    min = max = sum = x;

    i = 2;
    while ( i <= n )
    {
        printf("\nΔώσε μου τον όρο υπ' αριθ. %d: ", i);
        scanf("%f", &x);
        if(x < min) min = x;
        if(x > max) max = x;
        sum += x;
        i++;
    }
}

```

```

    }
    printf("\nΟ ελάχιστος είναι %f και ο μέγιστος %f", min, max );
    printf("\nΟ μέσος όρος είναι %f\n", sum/n);
}

```

Πρόγραμμα p3-4 (επίλυση προβλήματος)
 Το αποτέλεσμα γράφεται στο αρχείο beers.txt.

```

#include <stdio.h>

int main( )
{
    int i;
    FILE *myFile;

    myFile = fopen("beers.txt", "w");

    for(i = 99; i > 1; i--)
    {
        fprintf(myFile, "%d bottles of beer on the wall, %d bottles of
beer.\n", i, i);
        fprintf(myFile, "Take one down and pass it around, %d bottles of beer
on the wall.\n\n");
    }
    fprintf(myFile, "1 bottle of beer on the wall, 1 bottle of beer.\n");
    fprintf(myFile, "Take one down and pass it around, no more bottles of
beer on the wall.\n\n");
    fprintf(myFile, "No more bottles of beer on the wall, no more bottles of
beer.\n");
    fprintf(myFile, "Go to the store and buy some more, 99 bottles of beer on
the wall.\n");

    fclose(myFile);
}

```

Πρόγραμμα p3-5 (επίλυση προβλήματος)
 #include <stdio.h>

```

int main( )
{
    int i;
    FILE *myFile;

```

```

myFile = fopen("beers.txt", "w");

i = 99;
while( i > 1 )
{
    fprintf(myFile, "%d bottles of beer on the wall, %d bottles of
beer.\n", i, i);
    fprintf(myFile, "Take one down and pass it around, %d bottles of beer
on the wall.\n\n");
    i--;
}
fprintf(myFile, "1 bottle of beer on the wall, 1 bottle of beer.\n");
fprintf(myFile, "Take one down and pass it around, no more bottles of
beer on the wall.\n\n");
fprintf(myFile, "No more bottles of beer on the wall, no more bottles of
beer.\n");
fprintf(myFile, "Go to the store and buy some more, 99 bottles of beer on
the wall.\n");

fclose(myFile);
}

```

Πρόγραμμα p3-6 (βελτίωση προγράμματος)

Το πρόγραμμα προφανώς υπολογίζει την παράσταση

$$\frac{1}{x(x-1)(x-2)}$$

για δεδομένο x. Για να κάνει τη διαίρεση ελέγχει αν ο παρονομαστής είναι μηδέν βλέποντας αν το x παίρνει τιμή 0, 1 ή 2. Για να το φέρουμε σε μορφή πιο πυκνή αλλά και ευανάγνωστη συγχρόνως υπολογίζουμε πρώτα τον παρονομαστή $x(x-1)(x-2)$ και αξιοποιούμε το γεγονός ότι οι παραστάσεις έχουν και λογική τιμή (ΑΛΗΘΗΣ αν είναι διάφορη του μηδενός και ΨΕΥΔΗΣ αν είναι ίση με μηδέν).

```

#include <stdio.h>

int main( )
{
    float x, x1;

    printf("\nΔώσε μου έναν πραγματικό αριθμό: ");
    scanf("%f", &x);
    x1 = x*(x-1.0)*(x-2.0);
    if( x1 )
        printf("Το αποτέλεσμα είναι %f \n", 1.0/x1);
    else
        printf(" x = %f δίνει παρονομαστή μηδέν!\n", x);
}

```