Εισαγωγή στον Προγραμματισμό

Ενότητα C++ (75%) :

12η εβδομάδα

Παναγιώτης Τζουνάκης

(βασισμένο στις σημειώσεις που ευγενικά προσέφερε ο καθηγητής κ. Γεώργιος Ραχώνης)

Άνοιξη 2026

Παράδειγμα 1

Το παραγοντικό ενός μη αρνητικού ακέραιου αριθμού *n* ορίζεται με αναδρομικό τρόπο από τις σχέσεις:

$$0!=1 \quad \text{και} \quad n!=n(n-1)! \quad \text{για } n>0.$$

Να γράψετε πρόγραμμα που ζητάει από το χρήστη να εισάγει ένα μη αρνητικό ακέραιο αριθμό *n* και υπολογίζει και εκτυπώνει το παραγοντικό του.

```
#include<iostream>
using namespace std;
long int factorial(int);
int main ( )
{
int n;
long int fact;
cout<<“\nAssign value to n :”;
cin>>n;
while (n<0)
{
    cout<<“\n The number n should be > 0”;
    cout<<“\nAssign again value to n:”;
    cin>>n;
}
```

```
fact=factorial(n);
cout<<"\nThe factorial of n is : "<<fact;
return 0;
}
long int factorial(int k)
{
long int f;
if (k==0)
    f=1;
else
    f=k*factorial(k-1);
return(f);
}
```

Παράδειγμα 2

Να γράψετε πρόγραμμα που θα διαβάζει ένα 5x5 πίνακα *A* θετικών ακεραίων μικρότερων ή ίσων του 20 (απαιτείται έλεγχος ορθότητας δεδομένων) και θα δημιουργεί τον 5x5 πίνακα *B* του οποίου τα στοιχεία δίνονται από τον τύπο

$$B_{ij} = \sum_{1 \le k \le j} A_{ik} \; !$$

για κάθε 1≤i,j≤5.

Ο υπολογισμός των παραγοντικών θα γίνεται με τη χρήση συνάρτησης. Το πρόγραμμα θα τυπώνει και τους δύο πίνακες.

```
#include<iostream>
using namespace std;
long int factorial(int);
int main()
{
    int A[5][5], i,j,k;
    long int B[5][5];
    for (i=0;i<5;i++)
    {
        for (j=0;j<5;j++)
        {
            do
            {
                cout<<"\nAssign a value greater than 0
                    and lower than 21 to the ("<<i+1<<","
                     <<j+1<<") element of A :";
                cin>>A[i][j];
            }
            while(A[i][j]<0 || A[i][j]>20);
        }
    }
```

```
for (i=0;i<5;i++)
{
        for (j=0;j<5;j++)
        {
                B[i][j]=0;
                for (k=0;k<=j;k++)
                        B[i][j]=B[i][j]+factorial(A[i][k]);
         }
 }
 cout<<"\n The matrix A  \n\n\n";
 for (i=0;i<5;i++)
{
        cout<<endl;
        for (j=0;j<5;j++)
                cout<<A[i][j]<<"        ";
 }
```

```
    cout<<"\n \n\n The matrix B  \n\n\n";
    for (i=0;i<5;i++)
    {
        cout<<endl;
        for (j=0;j<5;j++)
            cout<<B[i][j]<<"        ";
    }
    return 0;
}
long int factorial(int m)
{
    long int f;
    if (m==0)
        f=1;
    else
        f=m*factorial(m-1);
    return(f);
}
```

Παράδειγμα 3

Γράψτε πρόγραμμα που να εκτυπώνει το αποτέλεσμα υπολογισμού μιας από ένα σύνολο συναρτήσεων (τετράγωνο, κύβος, ύψωση σε δύναμη, αντίστροφος, λογάριθμος, απόλυτη τιμή) της μεταβλητής εισόδου x, κατόπιν αντίστοιχης επιλογής του χρήστη.

```
#include<iostream>
#include<math.h>
using namespace std;
int main()
{
        int choice, n;
        float x, y;

        cout<<"\n enter value of x:";
        cin>>x;
        cout<<"Choose function to calculate:\n";
        cout<<"Enter 1 for square of x,\n";
        cout<<"Enter 2 for cube of x,\n";
        cout<<"Enter 3 for power of x,\n";
        cout<<"Enter 4 for inverse of x,\n";
```

```
cout<<"Enter 5 for logarithm of x,\n";
cout<<"Enter 6 for absolute value of x,\n";
cout<<"choice=";
cin>>choice;
if (choice==1)
        y=pow(x,2);
else
        if (choice==2)
                y=pow(x,3);
        else
                if (choice==3)
                {
                        cout<<"Enter exponent:";
                        cin>>n;
                        cout<<"\n";
```

```
                y=pow(x,n);
            }
            else
                if (choice==4)
                {
                    if (x!=(0.0))
                        y=1.0/x;
                    else
                    {
                        cout<<"Can not
calculate inverse of zero!\n";
                        return -1;
                    }
                }
                else
```

```
if (choice==5)
{
        if (x>(0.0))
                y=log(x);
        else
        {
                cout<<"Can not calculate logarithm of non positive number!\n";
                return -1;
        }
}
else
        if (choice==6)
                y=fabs(x);
        else
```

```
                                                {

        cout<<"invalid choice "<<choice<<", exiting without
calculation\n";
                                                        return -1;
                                                }
        cout<<"result y="<<y<<"\n";
        return 0;
}
```

Επιπλέον προτάσεις-εντολές ελέγχου ροής προγράμματος

- Switch
- Βοηθητικές εντολές:
  - –break
  - –continue
  - –goto

Εντολή switch

Υλοποιεί εκτέλεση κώδικα «υπό συνθήκη» αναλόγως της τιμής μιας έκφρασης ελέγχου

```
switch (expression) {
case const expr 1 :
statements 1
case const expr 2 :
statements 2
...
case const expr n :
statements n
default:
statements
}
```

- Η έκφραση ελέγχου πρέπει να είναι ακέραιου ή διακριτού τύπου (ή δυνάμει διακριτού τύπου π.χ. Bool
- Οι εκφράσεις κάθε περίπτωσης (case) πρέπει να είναι σταθερές (να επιδέχονται αποτίμησης κατά το χρόνο μεταγλώττισης).
- Το πρόγραμμα εκτελεί τις εντολές της περίπτωσης που αντιστοιχεί στην έκφραση ελέγχου
- ΚΑΙ ΣΥΝΕΧΙΖΕΙ ΣΤΗΝ ΕΠΟΜΕΝΗ περίπτωση, εκτός κι αν υπάρχει εντολή αλλαγής ροής (break, goto, return, …)

Εντολή break

Χρησιμοποιείται **μόνο** α) με την εντολή switch β) εντός μπλοκ εντολών βρόγχου (loop) for, while, do-while

Ο έλεγχος ροής προγράμματος μεταφέρεται στην επόμενη εντολή από αυτή που περιέχει την εντολή break.

Παράδειγμα break

```
…
int i;
while (true) {
cout << "Δώσε ένα θετικό ακέραιο: ";
cin >> i;
cout << '\n';
if (i > 0) {
break;
}
cout << “αρνητικός ακέραιος, δοκίμασε ξανά\n";
}
…
```

Εντολή continue

Χρησιμοποιείται **μόνο** εντός μπλοκ εντολών βρόγχου (loop) for, while, do-while

Ο έλεγχος ροής προγράμματος μεταφέρεται στο τέλος του μπλοκ που περιέχει την εντολή break και εκτελείται η επόμενη επανάληψη του μπλοκ (αν αυτό προβλέπεται από την αντίστοιχη τιμή δείκτη ή συνθήκη ελέγχου).

Παράδειγμα continue

```
…
int i, j=0;
for (i=0; i < 1000; ++i) {
if ((i %5) != 0) {
continue;
}
j=j+i;
cout << “i=“i<<“ j=“<<j<< “\n”;
}
…
```

Εντολή goto

Δηλώνουμε μια ετικέτα (label) η οποία αντιστοιχεί σε κάποια εντολή ως εξής

labelname: statement;

…

goto labelname;

ΣΗΜΕΙΩΣΗ: περιττή και δυνητικά επικίνδυνη εντολή, χρησιμοποιείτε μετά προσοχής!

Παράδειγμα 3Β

```
#include<iostream>
#include<math.h>
using namespace std;


int main()
{
        int choice, n;
        float x, y;
```

```
cout<<"\n enter value of x:";
cin>>x;
cout<<"Choose function to calculate:\n";
cout<<"Enter 1 for square of x,\n";
cout<<"Enter 2 for cube of x,\n";
cout<<"Enter 3 for power of x,\n";
cout<<"Enter 4 for inverse of x,\n";
cout<<"Enter 5 for logarithm of x,\n";
cout<<"Enter 6 for absolute value of x,\n";
cout<<"choice=";
cin>>choice;
```

```
switch (choice) {
        case 1:
                y=pow(x,2);
                break;
        case 2:
                y=pow(x,3);
                break;
        case 3:
                cout<<"Enter exponent:";
                cin>>n;
                cout<<"\n";
```

```
                y=pow(x,n);
            break;
        case 4:
            if (x!=(0.0))
                y=1.0/x;
            else
            {
                cout<<"Can not calculate
inverse of zero!\n";
                return -1;
            }
```

```
                    break;
            case 5:
                    if (x>(0.0))
                            y=log(x);
                    else
                    {
                            cout<<"Can not calculate
logarithm of non positive number!\n";
                            return -1;
                    }
                    break;
```

```
            case 6:
                y=fabs(x);
                break;
            default:
                cout<<"invalid choice
"<<choice<<", exiting without calculation\n";
                return -1;
        }
        cout<<"result y="<<y<<"\n";
        return 0;
}
```

Παράδειγμα 4

Να γράψετε πρόγραμμα που θα υπολογίζει με τη βοήθεια συναρτήσεων τον *n-οστό* όρο της αναδρομικής ακολουθίας

$x_n=(n-4)!x_{n-1}+(n-3)!x_{n-2}+(n-2)!x_{n-3}+(n-1)!x_{n-4}$ **για** $n>4$.

Οι 4 πρώτοι όροι x_1,x_2,x_3,x_4 **θα καθορίζονται από το χρήστη.**

```
#include<iostream>
using namespace std;
long int factorial(int);
long int rec(int, int, int, int, int);
int main()
{
    int x1,x2,x3,x4,n;
    long int x;
    cout<<"\nAssign value to the first term :";
    cin>>x1;
    cout<<"\nAssign value to the second term :";
    cin>>x2;
```

```
cout<<“\nAssign value to the third term :”;
cin>>x3;
cout<<“\nAssign value to the fourth term :”;
cin>>x4;
cout<<”\nAssign a positive value to n :”;
cin>>n;
while (n<=0)
{
      cout<<“\n The number n must be positive”;
      cout<<“\nAssign again value to n :”;
      cin>>n;
}
```

```
    x=rec(x1,x2,x3,x4,n);
    cout<<"\nThe "<<n<<"term of the
                                        sequence is :"<<x;
    return 0;
}
long int factorial(int k)
{
    long int f;
    if (k==0)
        f=1;
    else
        f=k*factorial(k-1);
    return(f);
}
```

```
long int rec(int y1, int y2, int y3, int y4, int m)
{
    long int y;
    if (m==1)
        y=y1;
    else if (m==2)
        y=y2;
    else if (m==3)
        y=y3;
    else if (m==4)
        y=y4;
```

```
    else
        y=factorial(m-4)*rec(y1,y2,y3,y4,m-1)+
          factorial(m-3)*rec(y1,y2,y3,y4,m-2)+
          factorial(m-2)*rec(y1,y2,y3,y4,m-3)+
          factorial(m-1)*rec(y1,y2,y3,y4,m-4);
    return(y);
}
```

Παράδειγμα 5

Να γράψετε πρόγραμμα που θα διαβάζει δύο 5x5 πίνακες Α και Β θετικών ακεραίων μικρότερων ή ίσων του 20 (απαιτείται έλεγχος ορθότητας δεδομένων) και θα δημιουργεί τον 5x5 πίνακα C του οποίου τα στοιχεία δίνονται από τον τύπο

$$c_{ij} = \sum_{1 \le k \le j} a_{ik}! + \sum_{j \le k \le 5} b_{ik}!$$

για κάθε $1 \le i, j \le 5$

Ο υπολογισμός των παραγοντικών θα γίνεται με τη χρήση συνάρτησης. Το πρόγραμμα θα τυπώνει και τους 3 πίνακες.

```
#include<iostream>
using namespace std;
long int factorial(int);
int main()
{
    int A[5][5],B[5][5], i,j,k;
     long int C[5][5];
```

```
for (i=0;i<5;i++)
    {
        for (j=0;j<5;j++)
        {
                do
                {
                        cout<<"\nAssign a value greater than 0
                            and lower than 21 to the
                                ("<<i+1<<","
                                <<j+1<<") element of A :";
                        cin>>A[i][j];
                }
                while(A[i][j]<=0 || A[i][j]>20);
         }
    }
```

```
for (i=0;i<5;i++)
    {
        for (j=0;j<5;j++)
        {
                do
                {
                        cout<<"\nAssign a value greater than 0
                            and lower than 21 to the
                                ("<<i+1<<","
                                <<j+1<<") element of B :";
                        cin>>B[i][j];
                }
                while(B[i][j]<=0 || B[i][j]>20);
         }
    }
```

```
for (i=0;i<5;i++)
{
      for (j=0;j<5;j++)
      {
            C[i][j]=0;
            for (k=0;k<=j;k++)
                  C[i][j]=C[i][j]+factorial(A[i][k]);
            for (k=j+1;k<5;k++)
                  C[i][j]=C[i][j]+factorial(B[i][k]);
       }
 }
```

```
cout<<"\n The matrix A  \n\n\n";
for (i=0;i<5;i++)
{
      cout<<endl;
      for (j=0;j<5;j++)
            cout<<A[i][j]<<"        ";
}
cout<<"\n \n\n The matrix B  \n\n\n";
for (i=0;i<5;i++)
{
      cout<<endl;
      for (j=0;j<5;j++)
            cout<<B[i][j]<<"        ";
}
```

```
    cout<<"\n \n\n The matrix C  \n\n\n";
    for (i=0;i<5;i++)
    {
        cout<<endl;
        for (j=0;j<5;j++)
            cout<<C[i][j]<<"        ";
    }
    return 0;
}
long int factorial(int m)
{
    long int f;
    if (m==0)
        f=1;
    else
        f=m*factorial(m-1);
    return(f);
}
```

Παράδειγμα 6

Να γράψετε πρόγραμμα που θα διαβάζει δύο 5x5 πίνακες Α και Β θετικών ακεραίων (απαιτείται έλεγχος ορθότητας δεδομένων) και θα δημιουργεί τον 5x5 πίνακα C του οποίου τα στοιχεία δίνονται από τον τύπο

$c_{ij} = \text{Μ.Κ.Δ.}(a_{ij} , b_{ij})$

για κάθε $1 \leq i, j \leq 5$.

Ο υπολογισμός των Μ.Κ.Δ. θα γίνεται με τη χρήση συνάρτησης. Το πρόγραμμα θα τυπώνει και τους 3 πίνακες.

```
#include<iostream>
using namespace std;
int mkd(int,int);
int main()
{
    int A[5][5],B[5][5],C[5][5],i,j;
```

```
for (i=0;i<5;i++)
    {
        for (j=0;j<5;j++)
        {
                do
                {
                    cout<<“\nAssign a value greater than 0
                     (“<<i+1<<”,” <<j+1<<“) element of A :”;
                     cin>>A[i][j];
                }
                while(A[i][j]<=0);
         }
    }
```

```
for (i=0;i<5;i++)
    {
        for (j=0;j<5;j++)
        {
                do
                {
                    cout<<“\nAssign a value greater than 0
                     (“<<i+1<<”,” <<j+1<<“) element of B :”;
                     cin>>B[i][j];
                }
                while(B[i][j]<=0);
         }
    }
```

```
for (i=0;i<5;i++)
{
      for (j=0;j<5;j++)
            C[i][j]=mkd(A[i][j],B[i][j]);
 }
cout<<"\n The matrix A  \n\n\n";
 for (i=0;i<5;i++)
{
      cout<<endl;
      for (j=0;j<5;j++)
            cout<<A[i][j]<<"        ";
 }
```

```
    cout<<"\n \n\n The matrix B  \n\n\n";
    for (i=0;i<5;i++)
   {
        cout<<endl;
        for (j=0;j<5;j++)
                cout<<B[i][j]<<"        ";
    }
   cout<<"\n \n\n The matrix C  \n\n\n";
    for (i=0;i<5;i++)
   {
        cout<<endl;
        for (j=0;j<5;j++)
                cout<<C[i][j]<<"        ";
    }
   return 0;
}
```

```
int mkd(int x, int y)
{
    int z;
    while(x!=y)
    {
        if(x>y)
                x=x-y;
        else
                y=y-x;
    }
    z=x;
    return(z);
}
```

Παράδειγμα 7

Να γράψετε πρόγραμμα που θα διαβάζει δύο 5x5 πίνακες Α και Β θετικών ακεραίων μικρότερων ή ίσων του 20 (απαιτείται έλεγχος ορθότητας δεδομένων) και θα δημιουργεί τον 5x5 πίνακα C του οποίου τα στοιχεία δίνονται από τον τύπο

$$c_{ij} = \text{Μ. Κ. Δ.}\left(\sum_{1\le k\le j} a_{ik}!\,, \sum_{j\le k\le 5} b_{ik}!\quad\right)$$

για κάθε $1 \le i, j \le 5$

Ο υπολογισμός των παραγοντικών και των Μ.Κ.Δ. θα γίνεται με τη χρήση συναρτήσεων. Το πρόγραμμα θα τυπώνει και τους 3 πίνακες.

```
#include<iostream>
using namespace std;
long int factorial(int);
long int mkd(long int, long int);
int main()
{
    int A[5][5],B[5][5], i,j,k;
    long int C[5][5],c1,c2;
```

```
for (i=0;i<5;i++)
    {
        for (j=0;j<5;j++)
        {
                do
                {
                        cout<<"\nAssign a value greater than 0
                            and lower than 21 to the
                                ("<<i+1<<","
                                <<j+1<<") element of A :";
                        cin>>A[i][j];
                }
                while(A[i][j]<=0 || A[i][j]>20);
         }
    }
```

```
for (i=0;i<5;i++)
    {
        for (j=0;j<5;j++)
        {
                do
                {
                        cout<<"\nAssign a value greater than 0
                            and lower than 21 to the
                        ("<<i+1<<","
                                <<j+1<<") element of B :";
                        cin>>B[i][j];
                }
                while(B[i][j]<=0 || B[i][j]>20);
         }
    }
```

```
for (i=0;i<5;i++)
{
      for (j=0;j<5;j++)
      {
            c1=0;
            c2=0;
            for (k=0;k<=j;k++)
                  c1=c1+factorial(A[i][k]);
            for (k=j;k<5;k++)
                  c2=c2+factorial(B[i][k]);
            C[i][j]=mkd(c1,c2);
      }
}
```

```
cout<<"\n The matrix A  \n\n\n";
for (i=0;i<5;i++)
{
      cout<<endl;
      for (j=0;j<5;j++)
            cout<<A[i][j]<<"        ";
}
cout<<"\n \n\n The matrix B  \n\n\n";
for (i=0;i<5;i++)
{
      cout<<endl;
      for (j=0;j<5;j++)
            cout<<B[i][j]<<"        ";
}
```

```
    cout<<"\n \n\n The matrix C  \n\n\n";
    for (i=0;i<5;i++)
    {
        cout<<endl;
        for (j=0;j<5;j++)
            cout<<C[i][j]<<"        ";
    }
    return 0;
}
long int factorial(int m)
{
    long int f;
    if (m==0)
        f=1;
    else
        f=m*factorial(m-1);
    return(f);
}
```

```
long int mkd(long int x, long int y)
{
    long int z;
    while(x!=y)
    {
        if(x>y)
                x=x-y;
        else
                y=y-x;
    }
    z=x;
    return(z);
}
```

Άσκηση 1

Να γράψετε πρόγραμμα που να υπολογίζει τον βαθμό του μαθήματος «Εισαγωγή στον προγραμματισμό» για τους 41 φοιτητές του τμήματος 2Γ. Οι 3 καλύτερες βαθμολογίες των 7 σετ ασκήσεων για το σπίτι μετρούν για το 30% του βαθμού. Το 1ο σετ βαθμολογείται από 0 ως 1 και τα υπόλοιπα 6 από 0 ως 0,5 μονάδες, όλα με ακρίβεια ως και 3 δεκαδικών ψηφίων. Όποιος δεν παραδίδει ένα σετ ασκήσεων βαθμολογείται με 0 σε αυτό το σετ. Η τελική εξέταση μετρά για το υπόλοιπο 70% του βαθμού. Στο τέλος ο βαθμός στρογγυλοποιείται σε ακέραιη μονάδα ή στο μισό της, όποιο είναι πιο κοντά. Σε κάθε εισαγωγή βαθμού γίνεται έλεγχος ορθότητας δεδομένων (σημ: ο χρήστης διαθέτει πληκτρολόγιο calculator μόνο με πλήκτρα 0..9 + - * / = . Enter)

Διαδικασία επίλυσης

1) Ανάλυση του προβλήματος σε απλούστερα
2) Αποσαφήνιση των σχέσεων:
   I. αριθμητικών : τι υπολογίζεται από τι
   II. Λογικών : τι προηγείται, τι έπεται, ποια μπορούν να τρέχουν παράλληλα

ανάμεσα στα απλούστερα προβλήματα

3) Αναδρομική εφαρμογή του παραπάνω κανόνα 1) σε κάθε απλούστερο πρόβλημα, μέχρι να είναι σαφής ο πρόδηλος τρόπος επίλυσης κάθε στοιχειώδους προβλήματος

Διαδικασία επίλυσης

1) Σύνθεση της υλοποίησης της λύσης με αντίστροφη σειρά.
2) Προγραμματισμός επιμέρους functions και μπλοκ κώδικα με αυτοτελείς υπολογισμούς, αποσφαλμάτωση, δοκιμές και έλεγχος των επιμέρους αποτελεσμάτων
3) Σύνθεση του συνολικού προγράμματος από τα επιμέρους τμήματα. Προσοχή στο πλήθος, στη συμβατότητα (τύπο) και στην ανάθεση ορθών τιμών στις παραμέτρους και στο εύρος (scope) χρήσης μεταβλητών από διάφορα κομμάτια του προγράμματος.

Άσκηση 1

Να γράψετε πρόγραμμα που να υπολογίζει τον βαθμό του μαθήματος «Εισαγωγή στον προγραμματισμό» για τους 41 φοιτητές του τμήματος 2Γ. Οι 3 καλύτερες βαθμολογίες των 7 σετ ασκήσεων για το σπίτι μετρούν για το 30% του βαθμού. Το 1ο σετ βαθμολογείται από 0 ως 1 και τα υπόλοιπα 6 από 0 ως 0,5 μονάδες, όλα με ακρίβεια ως και 3 δεκαδικών ψηφίων. Όποιος δεν παραδίδει ένα σετ ασκήσεων βαθμολογείται με 0 σε αυτό το σετ. Η τελική εξέταση μετρά για το υπόλοιπο 70% του βαθμού. Στο τέλος ο βαθμός στρογγυλοποιείται σε ακέραιη μονάδα ή στο μισό της, όποιο είναι πιο κοντά. Σε κάθε εισαγωγή βαθμού γίνεται έλεγχος ορθότητας δεδομένων (σημ: ο χρήστης διαθέτει πληκτρολόγιο calculator μόνο με πλήκτρα 0..9 + - * / = . Enter)

Εισαγωγή δεδομένων

Να γράψετε πρόγραμμα που να υπολογίζει τον βαθμό του μαθήματος «Εισαγωγή στον προγραμματισμό» για τους 41 φοιτητές του τμήματος 2Γ. Οι 3 καλύτερες βαθμολογίες των 7 σετ ασκήσεων για το σπίτι μετρούν για το 30% του βαθμού. Το 1ο σετ βαθμολογείται από 0 ως 1 και τα υπόλοιπα 6 από 0 ως 0,5 μονάδες, όλα με ακρίβεια ως και 3 δεκαδικών ψηφίων. Όποιος δεν παραδίδει ένα σετ ασκήσεων βαθμολογείται με 0 σε αυτό το σετ. Η τελική εξέταση μετρά για το υπόλοιπο 70% του βαθμού. Στο τέλος ο βαθμός στρογγυλοποιείται σε ακέραιη μονάδα ή στο μισό της, όποιο είναι πιο κοντά. Σε κάθε εισαγωγή βαθμού γίνεται έλεγχος ορθότητας δεδομένων (σημ: ο χρήστης διαθέτει πληκτρολόγιο calculator μόνο με πλήκτρα 0..9 + - * / = . Enter)

Επεξεργασία δεδομένων

…για τους 41 φοιτητές του τμήματος 2Γ. Οι 3 καλύτερες βαθμολογίες των 7 σετ ασκήσεων για το σπίτι μετρούν για το 30% του βαθμού. Το 1ο σετ βαθμολογείται από 0 ως 1 και τα υπόλοιπα 6 από 0 ως 0,5 μονάδες, όλα με ακρίβεια ως και 3 δεκαδικών ψηφίων. Όποιος δεν παραδίδει ένα σετ ασκήσεων βαθμολογείται με 0 σε αυτό το σετ. …

Ταξινόμηση στήλης πίνακα με 7 βαθμούς. Απαιτείται για καθένα από τους 41 φοιτητές (άρα θέλω πίνακα βαθμολογιών ασκήσεων 41 x 7.)

Προσθέτω τις βαθμολογίες στις 3 πρώτες θέσεις του (ταξινομημένου πλέον) πίνακα. Πολλαπλασιάζω το αποτέλεσμα x 0,3

Εισαγωγή δεδομένων

..Το 1ο σετ βαθμολογείται από 0 ως 1 και τα υπόλοιπα 6 από 0 ως 0,5 μονάδες, όλα με ακρίβεια ως και 3 δεκαδικών ψηφίων. Όποιος δεν παραδίδει ένα σετ ασκήσεων βαθμολογείται με 0 σε αυτό το σετ. … Σε κάθε εισαγωγή βαθμού γίνεται έλεγχος ορθότητας δεδομένων (σημ: ο χρήστης διαθέτει πληκτρολόγιο calculator μόνο με πλήκτρα 0..9 + - * / = . Enter)

Επιλέγω μεταβλητή τύπου float

Εισάγω αρχικές τιμές 0 στον πίνακα βαθμολογιών

Ελέγχω if (InpVar <0) || (InVar>1 && set ==1)||(InVar>0.5 && set >1)) get input again...