1) Write down code or PSEUDO algorithm to find out smallest and largest

amongst 3 elements.You can write program in any language you know (C/C++/Java/Python)

--------------------------------------------------------------------------------------

C Language:

#include<stdio.h>

int main()

{

int a,b,c;

scanf("%d%d%d",&a,&b,&c);

if(a > b && a > c)

{

printf("%d is largest",a);

}

else if(b > a && b > c)

{

printf("%d is largest",b);

}

else

{

printf("%d is largest",c);

}

if(a < b && a < c)

{

printf("%d is smallest",a);

}

else if(b < a && b < c)

{

printf("%d is smallest",b);

}

else

{

printf("%d is smallest",c);

}

}

--------------------------------------------------------------------------

C++ Language:

#include<bits/stdc++.h>

using namespace std;

int main()

{

int a,b,c;

cin>>a>>b>>c;

if(a > b && a > c)

{

cout<<a<<" is largest"<<endl;

}

else if(b > a && b > c)

{

cout<<b<<" is largest"<<endl;

}

else

{

cout<<c<<" is largest"<<endl;

}

if(a < b && a < c)

{

cout<<a<<" is smallest"<<endl;

}

else if(b < a && b < c)

{

cout<<b<<" is smallest"<<endl;

}

else

{

cout<<c<<" is smallest"<<endl;

}

}

---------------------------------------------------------------------------

Python Language:

a=int(input("enter the number")).strip()

b=int(input("enter the number")).strip()

c=int(input("enter the number")).strip()

if a > b and a > c:

print("a is largest")

elif b > a and b > c:

print("b is largest")

else:

print("c is largest")

if a < b and a < c:

print("a is smallest")

elif b < a and b < c:

print("b is smallest")

else:

print("c is smallest")

2) Write down code or PSEUDO algorithm to find out roots of a quadratic

equation. Please note that roots can be imaginary. You can write program

in any language you know (C/C++/Java/Python) OR even you can write PSEUDO

Code.

--------------------------------------------------------------------------

C Language:

#include<stdio.h>

int main()

{

int a,b,c,r1,r2,d;

printf (“enter the values of a b c”);

scanf (“ %d %d %d”, &a, &b, &c);

d= b\*b – 4\*a\*c;

if (d>0){

r1 = -b+sqrt (d) / (2\*a);

r2 = -b-sqrt (d) / (2\*a);

printf (“The real roots = %f %f”, r1, r2);

}

else if (d==0){

r1 = -b/(2\*a);

r2 = -b/(2\*a);

printf (“roots are equal =%f %f”, r1, r2);

}

else{

printf(“Roots are imaginary”);

}

}

--------------------------------------------------------------------------

C++ Language:

#include<bits/stdc++.h>

using namespace std:

int main()

{

int a,b,c,r1,r2,d;

cin>>a>>b>>c;

d=b\*b-4\*a\*c;

if(d>0)

{

r1=-b+sqrt(d)/(2\*a);

r2=-b-sqrt(d)/(2\*a);

cout<<"The real roots are ="<<r1<<r2<<endl;

}

else if(d==0)

{

r1=-b/(2\*a);

r2=-b/(2\*a);

cout<<"The roots are equal ="<<r1<<r2<<endl;

}

else

{

cout<<"Roots are imaginary";

}

}

---------------------------------------------------------------------------

Python Language:

a=int(input(" ")).strip()

b=int(input(" ")).strip()

c=int(input(" ")).strip()

d=b\*b-4\*a\*c;

if d>0:

r1=-b+sqrt(d)/(2\*a);

r2=-b-sqrt(d)/(2\*a);

print("The real roots are =" r1 r2)

elif d==0:

r1=-b/(2\*a);

r2=-b/(2\*a);

print("The roots are equal =" r1 r2)

else:

print("Roots are imaginary")

a) readMatrix function

Function should read number of rows, columns and elements of Matrix

void read(int r,int c,int a[][])

{

for(int i=0;i<r;i++)

{

for(int j=0;j<c;j++)

{

cin>>a[i][j];

}

}

}

--------------------------------------------------------------------------

b) displayMatrix function

Function should display a particular matrix

int display(int r,int c,int a[][])

{

for(int i=0;i<r;i++)

{

for(int j=0;j<c;j++)

{

cout<<a[i][j];

}

}

}

---------------------------------------------------------------------------

c) sumAllMatrixElemets function â€¦Function should sum of all elements of given matrix.

int sum(int r,int c,int a[][])

{

sum=0;

for(int i=0;i<r;i++)

{

for(int j=0;j<c;j++)

{

sum +=a[i][j];

}

}

return sum;

}

---------------------------------------------------------------------------

d) averageOfMatrix function â€¦Function should find out average of matrix elements

int avg(int r,int c,int a[][])

{

sum=0;

for(int i=0;i<r;i++)

{

for(int j=0;j<c;j++)

{

sum +=a[i][j];

}

}

int avg=sum/r\*c;

return avg;

}

----------------------------------------------------------------------------

b) AddMatrix function

Function should add two matrices to collect result in another matrix. (check if number of rows and columns are same)

int add(int r,int c,a[][],b[][],c[][])

{

for(int i=0;i<r;i++)

{

for(int j=0;j<c;j++)

{

cin>>a[i][j];

}

}

for(int i=0;i<r;i++)

{

for(int j=0;j<c;j++)

{

cin>>b[i][j];

}

}

for(int i=0;i<r;i++)

{

for(int j=0;j<c;j++)

{

c[i][j]=a[i][j]+b[i][j];

}

}

for(int i=0;i<r;i++)

{

for(int j=0;j<c;j++)

{

cout<<c[i][j];

}

}

}

---------------------------------------------------------------------------

4) Write PSEUDO algorithm to perform an encryption as well as decryption

of a file. Use any algorithm to perform above said activity.

Simplest possible algorithm is to read every character, add 128 to

it and stores it while performing an encryption and do reverse

step while performing a decryption. (You can even give a try to write

program in C/C++/Java/Python)

Your program should

a) Read a file character by character

b) Encrypt a file

d) Display contents of

#include <stdio.h>

int main()

{

int i, x;

char str[100];

printf(" enter a string");

gets(str);

for(i = 0; (i < 100 && str[i] != '\0'); i++)

str[i] = str[i] + 3;

printf("Encrypted string: %s", str);

for(i = 0; (i < 100 && str[i] != '\0'); i++)

str[i] = str[i] - 3;

printf("Decrypted string: %s", str);

}

5) Write down an PSEUDO algorithm or code in language you know

(C/C++/Java/Python) that reads in a mark of a student (which is an integer

between 0 and 100) and prints the corresponding grades (A-F).

The mark-to-grade conversion table is as follows:

Grade A B C D F

Range ≥80 65-79 50-64 40-49 <40

1. Modify the program so that it checks if the input is between 0 and 100.

If not, it should ask the user to input again until the input is in the

correct range. Use while statements. 2. Modify the program so that it

repeats the above computation on 50 students. Use while statements.

#include<bits/stdc++.h>

using namespace std;

int grade(int m)

{

if(m>=80 or m<=100)

{

cout<<"your grade is A";

}

else if(m<=79 or m>=65)

{

cout<<"your grade is B";

}

else if(m<=64 or m>=50)

{

cout<<"your grade is C";

}

else if(m<=49 or m>=40)

{

cout<<"your grade is D";

}

else

{

cout<<"your grade is F";

}

}

int main()

{

int m;

cin>>m;

int i=0;

while(i!=50)

{

if(m<=100 or m>=0)

{

grade(m);

i++;

}

else

{

cout<<"please enter the marks in range 0 to 100";

}

}

}