Institute of Computer Technology
B. Tech Computer Science and Engineering
Subject: ESFP-II (2CSE203)

PRACTICAL-3

AIM: - To learn about branching, looping and logical operators in C++.

1. Determines a student's grade make a program which will read three types of scores (quiz, mid-term, and final scores) and determine the grade based on the following rules:
-if the average score =90% =>grade=A
-if the average score >= 70% and <90% => grade=B
-if the average score>=50% and <70% =>grade=C
-if the average score<50% =>grade=F

CODE:

```
#include <iostream>
using namespace std;
int main() {
   int quiz,mid_term,final,avg;

   cout<<"\nEnter Marks of Quiz: ";
   cin>>quiz;
   cout<<"\nEnter Marks of Mid-Term: ";
   cin>>mid_term;
   cout<<"\nEnter Final Score: ";
   cin>>final;

   avg=(quiz+mid_term+final)/3;

   if (avg>=90)
   {
      cout<<"\nGrade-A";
   }
   else if (avg<90 && avg>=70)
   {
      cout<<"\nGrade-B";
   }
   else if (avg<70 && avg>=50)
   {
      cout<<"\nGrade-C";
   }
   else
   {
      cout<<"\nGrade-F";
   }

   return 0;
}
```

OUTPUT:

```
Enter Marks of Quiz: 99

Enter Marks of Mid-Term: 99

Enter Final Score: 99

Grade-A
PS C:\Users\admin\Google Drive\B-Tech\SEM-2\ESFP-2\ESFP-Practicals\Prac-3>
```

```
PS C:\Users\admin\Google Drive\B-Tech\SEM-2\ESFP-2\ESFP-Practicals\Prac-3> cd "c:\Users\admi

Enter Marks of Quiz: 75

Enter Marks of Mid-Term: 69

Enter Final Score: 80

Grade-B
PS C:\Users\admin\Google Drive\B-Tech\SEM-2\ESFP-2\ESFP-Practicals\Prac-3>
```

```
Enter Marks of Quiz: 60

Enter Marks of Mid-Term: 60

Enter Final Score: 60

Grade-C
PS C:\Users\admin\Google Drive\B-Tech\SEM-2\ESFP-2\ESFP-Practicals\Prac-3>
```

```
Enter Marks of Quiz: 40

Enter Marks of Mid-Term: 35

Enter Final Score: 42

Grade-F
PS C:\Users\admin\Google Drive\B-Tech\SEM-2\ESFP-2\ESFP-Practicals\Prac-3>
```

---

2. Compute the real roots of the equation: ax2+bx+c=0.

The program will prompt the user to input the values of a, b, and c. It then computes the real roots of the equation based on the following rules:

-if a and b are zero=> no solution

-if a is zero=>one root (-c/b)

-if b2-4ac is negative=>no roots

-Otherwise=> two roots

The roots can be computed using the following formula: $x1=-b+(b2-4ac)^{1/2}/2a$ $x=-b-(b2-4ac)^{1/2}/2a$

CODE:

```
#include <iostream>
#include <cmath>
using namespace std;
```

```
int main() {
   float a,b,c,root,delta,x,x1,r,i;

   cout<<"\nEnter co-efficient of x^2: ";
   cin>>a;
   cout<<"\nEnter co-efficient of x: ";
   cin>>b;
   cout<<"\nEnter value for c: ";
   cin>>c;

   if (a==0)
   {
      cout<<"\nThis is not a quadratic equation.";
      return 0;
   }
   else if (a==0 && b==0)
   {
      cout<<"\nThis is not a quadratic equation.";
      return 0;
   }

   cout<<"\nThe equation is "<<"("<<a<<")"<<"x^2+"<<"("<<b<<")"<<"x+"<<"("<<c<<")";

   delta=((b*b)-4*a*c);
   if (delta>=0)
   {
      if (delta==0)
      {
         x1=x=(-b/2*a);
         cout<<"\nReal and same roots: "<<x1<<" & "<<x;
      }
      else {
         x,x1=0;
         x1=(((-b)+ sqrt(delta))/(2*a));
         x=(((-b)- sqrt(delta))/(2*a));
         cout<<"\nEquation has two different real roots: -";
         cout<<"\n1st Root: "<<x1;
         cout<<"\n2nd Root: "<<x;
      }
   }
   if (delta<0)
   {
      r=((-b)/2*a);
      i= (sqrt(-delta)/(2*a));
      cout<<"\nThis equation has two imaginary roots";
      cout<<"\nRoot 1: "<<r<<" + "<<i<<"i";
      cout<<"\nRoot 2: "<<r<<" - "<<i<<"i";
   }

   return 0;
}
```

OUTPUT:

```
Enter co-efficient of x^2: 1

Enter co-efficient of x: 5

Enter value for c: 6

The equation is (1)x^2+(5)x+(6)
Equation has two different real roots: -
1st Root: -2
2nd Root: -3
PS C:\Users\admin\Google Drive\B-Tech\SEM-2\ESFP-2\ESFP-Practicals\Prac-3>
```

```
Enter co-efficient of x^2: 2

Enter co-efficient of x: 3

Enter value for c: 4

The equation is (2)x^2+(3)x+(4)
This equation has two imaginary roots
Root 1: -3 + 1.19896i
Root 2: -3 - 1.19896i
PS C:\Users\admin\Google Drive\B-Tech\SEM-2\ESFP-2\ESFP-Practicals\Prac-3>
```

3. Follow the given pattern and write a code for it.

```
*
***
*****
*******
*********
*********
*******
*****
***
*
```

CODE:

```
#include<iostream>
using namespace std;
int main()
{
   int n, i , j;
   n=9;
   for(i = 1; i <= n; i=i+2) {
      for(j = 1; j <= i; j++) {
         cout << "*";
      }
      cout<<"\n";
   }
   for(i = n; i >= 1; i=i-2) {
      for(j = 1; j <= i; j++) {
         cout << "*" ;
```

```
        }
        cout<<"\n";
    }
    return 0;
}
```

OUTPUT:

```
  *
  ***
  *****
  *******
  *********
  *********
  *******
  *****
  ***
  *
  PS C:\Users\admin\Google Drive\B-Tech\SEM-2\ESFP-2\ESFP-Practicals\Prac-3>
```

4. Take n positive numbers as a user input. The program will terminate if one of those number is not positive.

CODE:

```
#include <iostream>
using namespace std;
int main() {
    int n;
    for(;;) {
        cout<<"Enter any positive number: ";
        cin>>n;
        if (n<0)
        {
            cout<<"This program will not accept any negative number...";
            break;
        }
    }
}
```

OUTPUT:

```
Enter any positive number: 4
Enter any positive number: 5
Enter any positive number: 8
Enter any positive number: 9
Enter any positive number: 6
Enter any positive number: 3
Enter any positive number: 77
Enter any positive number: -1
This program will not accept any negative number...
PS C:\Users\admin\Google Drive\B-Tech\SEM-2\ESFP-2\ESFP-Practicals\Prac-3>
```