UNIVERSITY OF CALCUTTA

BSC.(H) PRACTICAL EXAMINATION
(CBCS SYSTEM)
YEAR: 2022

Roll no:-213017-21-0038

Registration no:-017-1111-0566-21

Semester:-III

Paper code:-CMS-A-CC-3-6-PR

Date of examination:_____

INDEX

PROBLEM STATEMENT

1a) Write a program in C that will take a floating-point number as input. Show the integer part of that number.

Sample Input: 205.1
Output: 205

PROGRAM CODE:

```
#include<stdio.h>

int main()
{
        float a;
        printf("\n Enter the float number : ");
        scanf("%f",&a);
        printf("The Output is  :  %d",(int)a);
        return 0;
}
```

OUTPUT:

```
 Enter the float number : 2.565
The Output is  :  2
--------------------------------
Process exited after 7.419 seconds with return value 0
Press any key to continue . . .
```

PROBLEM STATEMENT

1b) Write a program in C that will take an integer as input. The input may be positive or negative. Display the number without the sign.

Sample input: -20

Output: 20

PROGRAM CODE:

```
#include<stdio.h>
#include<math.h>
int main()
{
        int a;
        printf("Enter an integer(negative or positive) :  ");
        scanf("%d",&a);
        printf("\nThe integer is %d ",abs(a));
   return 0;
}
```

OUTPUT :

```
Enter an integer(negative or positive) :  -6

The integer is 6
--------------------------------
Process exited after 2.805 seconds with return value 0
Press any key to continue . . .
```

```
Enter an integer(negative or positive) :  5

The integer is 5
--------------------------------
Process exited after 1.709 seconds with return value 0
Press any key to continue . . .
```

PROBLEM STATEMENT

1c) Write a program in C that will take a floating-point number as input. Display the number as its closest integer (lower value). Display the resultant number as integer and also as floating-point number.

Sample Input: 12.11
Output:12.00000

PROGRAM CODE:

```
#include<stdio.h>

int main()
{
        float a;
        printf("\n Enter the float number : ");
        scanf("%f",&a);
        printf("The Output is  :  %f",(float)(int)a);
        return 0;
}
```

OUTPUT :

```
 Enter the float number : 2.5000
The Output is  :  2.000000
--------------------------------
Process exited after 3.746 seconds with return value 0
Press any key to continue . . .
```

PROBLEM STATEMENT

1d) Write a program in C that will take a floating-point number as input. Display the number as its closest integer (upper value). Display the resultant number as integer and also as floating-point number.

Sample Input: 12.11
Output:13.00000

PROGRAM CODE :

```
#include<stdio.h>

int main()
{
	float a;
	printf("\n Enter the float number : ");
	scanf("%f",&a);
	printf("\nThe output : %f",(float)(int)a+1);
	return 0;
}
```

OUTPUT:

```
 Enter the float number : 2.111000

The output : 3.000000
--------------------------------
Process exited after 4.625 seconds with return value 0
Press any key to continue . . .
```

PROBLEM STATEMENT

1e) Write a program in C that will take a floating-point number as input. Display the number as its closest integer. Display the resultant number as integer and as floating-point number.

Sample Input: 6.9
Integer Output: 7
Floating point output: 7.0

PROGRAM CODE:

```
#include<stdio.h>
#include<math.h>

int main()
{
   float n;
   printf("Enter any  float number  : ");
   scanf("%f",&n);
   printf("%d",(int)round(n));
   printf("\n");
   printf("%f",round(n));

   return 0;
}
```

OUTPUT:

```
Enter any  float number  : 2.53
3
3.000000
PS C:\Users\user\Desktop\cc6 lab\New folder> cd "c:\Use
Enter any  float number  : 2.36
2
2.000000
PS C:\Users\user\Desktop\cc6 lab\New folder>
```

PROBLEM STATEMENT

2. a.Write a program in C to fetch two arrays (x and y) from the user. Then implement the Forward difference table.

x	0	1	2	3	4	5
y=f(x)	12	15	20	27	39	52

SOURCE CODE :

```
#include<stdio.h>

void display(float arr[][20],int n) //display function for displaying table
{
   int i,j;
   for(i=0;i<n;i++)
   {
      for(j=0;j<n-1-i;j++)
      {
         printf("%.2f    |",arr[i][j]); //printing elements
      }
      printf("\n");
   }
}
void forward_difference(float ar1[],float arr[],int n) // calling forward difference table
{
    float ar[20],arr1[20][20]={0}; //taking 2 D array to store elements
    int i,j;
      int c;
      for(i=0;i<n;i++)
      {
         arr1[i][0]=ar1[i];  //for  putting table contents
      }
      for(i=0;i<n;i++)
      {
         arr1[i][1]=arr[i];
      }
```

```
        for(i=(n-1),c=2;i>0;i--,c++)
        {
           for(j=0;j<i;j++)
           {
              arr1[j][c]=arr[j]=(arr[j+1]-arr[j]); //for calculating forward difference
           }

        }
        display(arr1,n+2); //calling display function

}
int main()
{
   int j,n,n1;
   float arr1[10],arr2[10];
   printf("Enter the  size  of  x: ");
   scanf("%d",&n);
   for(j=0;j<n;j++)
      {
         printf("Enter the data : "); // entering values of x
         scanf("%f",&arr1[j]);
      }
   printf("Enter the  size of y=f(x): ");
   scanf("%d",&n1);
   for(j=0;j<n1;j++)
      {
         printf("Enter the data : "); //enetering the values of y
         scanf("%f",&arr2[j]);
      }
   printf("\nx\t|y\t|y0\t|y1\t|y2\t|y3\t|y4\t|y5\n");
   printf("-------------------------------------------------------------------------\n");
    forward_difference(arr1,arr2,n1); // calling forward difference function

return 0;
```

}

OUTPUT:

```
PS C:\Users\user\Desktop\cc6 lab> cd "c:\Users\user\Desktop\cc6 lab\" ; if ($?) { g
nce.c -o Forward_difference } ; if ($?) { .\Forward_difference }
Enter the  size  of  x: 6
Enter the data : 0
Enter the data : 1
Enter the data : 2
Enter the data : 3
Enter the data : 4
Enter the data : 5
Enter the  size of y=f(x): 6
Enter the data : 12
Enter the data : 15
Enter the data : 20
Enter the data : 27
Enter the data : 39
Enter the data : 52

x       |y      |y0     |y1     |y2     |y3     |y4     |y5
-------------------------------------------------------------------------
0.00    |12.00  |3.00   |2.00   |0.00   |3.00   |-10.00 |
1.00    |15.00  |5.00   |2.00   |3.00   |-7.00  |
2.00    |20.00  |7.00   |5.00   |-4.00  |
3.00    |27.00  |12.00  |1.00   |
4.00    |39.00  |13.00  |
5.00    |52.00  |
0.00    |

PS C:\Users\user\Desktop\cc6 lab>
```

PROBLEM STATEMENT

b. Write a program in C to implement the Backward Difference Table.

PROGRAM CODE :

```
#include<stdio.h>

void display(int arr[][20],int n) //function for display of backward difference table
{
   int i,j;
   for(i=0;i<n-2;i++)
   {
      for(j=0;j<=i+1;j++)
      {
         printf("%d  |",arr[i][j]); //printing the elements of 2d array
      }
      printf("\n");
   }
}
void backward_difference(int ar1[],int arr[],int n)
{
    int i,ar[20],arr1[20][20]={0};
    int j;
      int c;
      for(i=0;i<n;i++)
      {
         arr1[i][0]=ar1[i]; //storing elements of x in the 2d array
      }
      for(i=0;i<n;i++)
      {
         arr1[i][1]=arr[i]; //same for y=f(x)
      }

         for(i=0,c=2;i<n;i++,c++)
         {
            for(j=n-1;j>i-1;j--)
```

```
            {
               arr1[j][c]=arr[j]=(arr[j]-arr[j-1]); //calculating backward difference
            }
         }
      display(arr1,n+2); // calling function for display

}
int main()
{
   int j,n,n1;
   int arr1[10],arr2[10];
   printf("Enter the  size : ");
   scanf("%d",&n);
   for(j=0;j<n;j++)
      {
         printf("Enter the data : "); //taking input for x
         scanf("%d",&arr1[j]);
      }
   printf("Enter the  size : ");
   scanf("%d",&n1);
   for(j=0;j<n1;j++)
      {
         printf("Enter the data : "); //taking input for y=f(x)
         scanf("%d",&arr2[j]);
      }
   printf("\n\tBACKWARD DIFFERENCE TABLE \n");
   printf("\nx\t|y\t|y0\t|y1\t|y2\t|y3\t|y4\t|y5\n");
   printf("-------------------------------------------------------------------------\n");
    backward_difference(arr1,arr2,n1); //backward difference table

return 0;
}
```

OUTPUT :

```
PS C:\Users\user\Desktop\cc6 lab> cd "c:\Users\user\Desktop\cc6 lab\" ; i
rd_difference } ; if ($?) { .\backward_difference }
Enter the  size : 6
Enter the data : 0
Enter the data : 1
Enter the data : 2
Enter the data : 3
Enter the data : 4
Enter the data : 5
Enter the  size : 6
Enter the data : 12
Enter the data : 15
Enter the data : 20
Enter the data : 27
Enter the data : 39
Enter the data : 52

        BACKWARD DIFFERENCE TABLE

x       |y      |y0     |y1     |y2     |y3     |y4     |y5
------------------------------------------------------------------------
0       |12     |
1       |15     |3      |
2       |20     |5      |2      |
3       |27     |7      |2      |0      |
4       |39     |12     |5      |3      |3      |
5       |52     |13     |1      |-4     |-7     |-10    |
PS C:\Users\user\Desktop\cc6 lab>
```

PROBLEM STATEMENT

3. Write a program in C to find out the value of f(x) and hence find f(6).

x	0	1	2	3	4	5
y=f(x)	41	43	47	53	61	71

PROGRAM CODE:

**** for calculating this question we are using lagrange's interpolation formula**

```
#include<stdio.h>

int main()
{
   int n,n1,i,j;
   float x[10],y[10],lag[10][10],X,Dr[10],Ydr[10],prod=1,sum;
   printf("\n\t Enter the size :  ");
   scanf("%d",&n);
   for(i=0;i<n;i++)
   {
      printf("Enter the value : "); //entering the value of the table
      scanf("%f",&x[i]);
   }
   printf("\n\t Enter the size :  ");
   scanf("%d",&n1);
   for(i=0;i<n1;i++)
   {
      printf("Enter the value : ");
      scanf("%f",&y[i]);
   }
   printf("\n\tEnter the value of x to find  f(x) : ");
   scanf("%f",&X);
   printf("|A\t|B\t|C\t|D\t|E\t|F\t\t|Dr\t|(Yr/Dr)");

printf("\n==========================================================================
==========");
```

```
for(i=0;i<n;i++)
{
   for(j=0;j<n;j++)
   {
      if(i==j)
         lag[i][j]= X-x[j]; //calculating the  difference
      else
         lag[i][j]= x[i]-x[0+j];
   }
}

for(i=0;i<n;i++)
{
   for(j=0;j<n;j++)
   {
      prod = prod * lag[i][j]; //calculating the row product
   }
   Dr[i] = prod;
   Ydr[i]=(y[i]/Dr[i]); //calculating yr /dr
   prod = 1;
}
printf("\n");
for(i=0;i<n;i++)
{
   for(j=0;j<n;j++)
   {
      printf("%.2f\t",lag[i][j]); //printing the table values
   }
   printf("\t%.2f\t %.2f",Dr[i],Ydr[i]); // printing d r  and yr/dr
   printf("\n");
}
sum=0;
for(i=0;i<n;i++)
{
```

```
    sum=sum+Ydr[i];
  }
  prod=1;
  for(i=0;i<n;i++)
  {
    for(j=0;j<n;j++)
    {
      if(i==j)
      {
        prod =  prod * lag[i][j]; // calculating the answer
      }
    }
  }

printf("\n=============================================================
============");
  printf("\n\n\t The value of f(%.2f) is  ====  %f ",X,(prod*sum)); //printing the answer

printf("\n=============================================================
============");
  return 0;
}
```

OUTPUT:

```
PS C:\Users\user\Desktop\cc6 lab> cd "c:\Users\user\Desktop\cc6 lab\" ; if ($?)
granges_interpolation } ; if ($?) { .\Lagranges_interpolation }

         Enter the size :  6
Enter the value : 0
Enter the value : 1
Enter the value : 2
Enter the value : 3
Enter the value : 4
Enter the value : 5

         Enter the size :  6
Enter the value : 41
Enter the value : 43
Enter the value : 47
Enter the value : 53
Enter the value : 61
Enter the value : 71

        Enter the value of x to find  f(x) : 6
|A      |B      |C      |D      |E      |F              |Dr     |(Yr/Dr)
================================================================================
6.00    -1.00   -2.00   -3.00   -4.00   -5.00           -720.00  -0.06
1.00    5.00    -1.00   -2.00   -3.00   -4.00           120.00   0.36
2.00    1.00    4.00    -1.00   -2.00   -3.00           -48.00   -0.98
3.00    2.00    1.00    3.00    -1.00   -2.00           36.00    1.47
4.00    3.00    2.00    1.00    2.00    -1.00           -48.00   -1.27
5.00    4.00    3.00    2.00    1.00    1.00            120.00   0.59

================================================================================

         The value of f(6.00) is  ====  82.999907
================================================================================
PS C:\Users\user\Desktop\cc6 lab>
```

PROBLEM STATEMENT

4. Write a program in C to evaluate the value of f(0.5) and f(2.8) using Newton's Forward/Backward Interpolation Formula:

x	0	1	2	3
y=f(x)	1	2	11	34

PROGRAM CODE:

```
#include<stdio.h>
#include<stdlib.h>

int main()
{
   int i, n, j, ch;
   double a[100][100], u, x, y, factorial, u1;
   printf("\nEnter the number of values: ");
   scanf("%d",&n);
   printf("\nEnter the values of x: \n");
   for(i=0;i<n;i++)
   {
      printf("Enter the value: ");
      scanf("%lf",&a[i][0]);
   }
   printf("\nEnter the values of f(x): \n");
   for(i=0;i<n;i++)
   {
      printf("Enter the value: ");
      scanf("%lf",&a[i][1]);
   }
   factorial=1;
   while(1)
   {
      printf("\n*********MENU*****************\n");
      printf("\n\t1.To use Newton's Forward interpolation");
```

```
    printf("\n\t2.To use Newton's Backward interpolation");
    printf("\n\t3.To exit");
    printf("\nEnter your choice..... : ");
    scanf("%d",&ch);
switch(ch)
{
    case 1:
    {
        printf("\nEnter the value x: ");
        scanf("%lf",&x);
        for(j=2;j<n+1;j++)
        {
            for(i=0;i<n-j+1;i++)
            {
                a[i][j] = a[i+1][j-1] - a[i][j-1];
            }
        }
        printf("\nThe Forward difference table is: \n");
        for(i=0;i<n;i++)
        {
            for(j=0;j<=n-i;j++)
            {
                printf("%0.3lf\t",a[i][j]);
            }
            printf("\n");
        }
        u=((x-a[0][0]))/((a[1][0] - a[0][0]));
        y=a[0][1];
        u1=u;
        for(i=2;i<=n;i++)
        {
            y = y + (u1 * a[0][i])/factorial;
            factorial = factorial * i;
            u1=u1*(u-(i-1));
```

```
    }
    break;
}
case 2:
{
    printf("\nEnter the value x: ");
    scanf("%lf",&x);
    for(j=2;j<n+1;j++)                          //Generating backward difference table
    {
        for(i=0;i<n-j+1;i++)
        {
            a[i][j] = a[i+1][j-1] - a[i][j-1];
        }
    }
    printf("\nThe Backward difference table is: \n");
    for(i=0;i<n;i++)
    {
        for(j=0;j<=n-i;j++)
        {
            printf("%0.3lf\t",a[i][j]);
        }
        printf("\n");
    }
    u=((x-a[n-1][0]))/((a[1][0] - a[0][0]));
    y=a[n-1][1];
    u1=u;
    j=2;            //column number
    for(i=n-2;i>=0;i--)
    {
        y = y + (u1 * a[i][j] )/factorial;
        factorial = factorial * j;
        u1=u1*(u+(j-1));
        j++;
    }
```

```
            break;
        }
        case 3:
        {
            exit(0);
            break;
        }
        default:
        {
            printf("\nInvalid Input.....!");
            break;
        }
    }
    printf("\n\nValue at x=%lf is = %lf",x,y);
    }
 return 0;
}
```

OUTPUT:

```
Enter the number of values: 4

Enter the values of x:
Enter the value: 0
Enter the value: 1
Enter the value: 2
Enter the value: 3

Enter the values of f(x):
Enter the value: 1
Enter the value: 2
Enter the value: 11
Enter the value: 34

*********MENU*****************

        1.To use Newton's Forward interpolation
        2.To use Newton's Backward interpolation
        3.To exit
Enter your choice..................... : 1

Enter the value x: 0.5

The Forward difference table is:
0.000   1.000   1.000   8.000   6.000
1.000   2.000   9.000   14.000
2.000   11.000  23.000
3.000   34.000


Value at x=0.500000 is = 0.875000
```

```
*********MENU*****************

        1.To use Newton's Forward interpolation
        2.To use Newton's Backward interpolation
        3.To exit
Enter your choice..................... : 2

Enter the value x: 2.8

The Backward difference table is:
0.000   1.000   1.000   8.000   6.000
1.000   2.000   9.000   14.000
2.000   11.000  23.000
3.000   34.000


Value at x=2.800000 is = 33.749667
*********MENU*****************

        1.To use Newton's Forward interpolation
        2.To use Newton's Backward interpolation
        3.To exit
Enter your choice..................... : 3
PS C:\Users\user\Desktop\cc6 lab>
```

PROBLEM STATEMENT

5. Write a program in C to determine one of the roots of the equation, x3-3x+1.06=0 by bisection method. Input the error tolerance as 0.001. Also print the number of operations.

PROGRAM CODE:

```
#include<stdio.h>

#include<conio.h>

#include<math.h>

#define f(x) (x * x * x - 3 * x + 1.06 )


int main()

{

        float a,b,c,res,res1,res2,tolerance;

                int i = 1;

        do

        {

                printf("\n\tEnter the value of a : ");


                scanf("%f",&a);


                res = f(a);


                printf("\n\t f(%g) := %g",a,res);


                printf("\n\tEnter the value of b : ");
```

```
        scanf("%f",&b);

        res1 = f(b);

        printf("\n\t f(%g) := %g",b,res1);
    }while(res * res1 > 0);

    printf("\n\t Enter the tolerance value  :  ");

    scanf("%f",&tolerance);

    printf("\n\t------------------------------------------------------------\n");

    printf("\n       N       f(a)              f(b)              (a+b)/2          f(new)");

    printf("\n\t------------------------------------------------------------\n");

    do
    {
    c = (a + b) / 2;  // here  average value  is  taken
    res = f(a);                    // calculating f(x) where x is the  first  value entered by
user
```

```
        res1 = f(b);            // calculating f(x) where x is the second value entered by user
        res2 = f(c);            // calculating f(x) where x is the average value calculated
        printf("\n\t%d\t%f\t%f\t%f\t%f",i,res,res1,c,res2);          //printing the values row wise
                if(res * res2 < 0)  // if the  product of the  "res " and  "res2" is  less than  zero
                {
                b = c;                  // then   b will be equal to the average value
                }
                else
                {
                a = c;                  // else  a  will be equal to the average value
                }
                i++;
        } while (fabs(f(c)) > tolerance);

        printf("\n\t-------------------------------------------------------------\n");

        printf("\n\nThe approximate root of the equation is :  %g",c);

        return 0;
}
```

OUTPUT:

```
        Enter the value of a : 0

         f(0) := 1.06
        Enter the value of b : 1

         f(1) := -0.94
         Enter the tolerance value  :  0.0001

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

        N       f(a)            f(b)            (a+b)/2         f(new)
        ------------------------------------------------------------

        1       1.060000        -0.940000       0.500000        -0.315000
        2       1.060000        -0.315000       0.250000        0.325625
        3       0.325625        -0.315000       0.375000        -0.012266
        4       0.325625        -0.012266       0.312500        0.153018
        5       0.153018        -0.012266       0.343750        0.069369
        6       0.069369        -0.012266       0.359375        0.028288
        7       0.028288        -0.012266       0.367188        0.007944
        8       0.007944        -0.012266       0.371094        -0.002178
        9       0.007944        -0.002178       0.369141        0.002879
        10      0.002879        -0.002178       0.370117        0.000350
        11      0.000350        -0.002178       0.370605        -0.000914
        12      0.000350        -0.000914       0.370361        -0.000282
        13      0.000350        -0.000282       0.370239        0.000034
        ------------------------------------------------------------


The approximate root of the equation is :  0.370239
--------------------------------
Process exited after 6.376 seconds with return value 0
Press any key to continue . . .
```

PROBLEM STATEMENT

6. Implement Lagrange's Interpolation table and evaluate the value of y for x = 10

x	5	6	9	11
y=f(x)	12	13	14	16

PROGRAM CODE:

```
#include<stdio.h>


int main()
{
   int n,n1,i,j;
   float x[10],y[10],lag[10][10],X,Dr[10],Ydr[10],prod=1,sum;
   printf("\n\t Enter the size :  ");
   scanf("%d",&n);
   for(i=0;i<n;i++)
   {
      printf("Enter the value : "); //entering the value of the table
      scanf("%f",&x[i]);
   }
   printf("\n\t Enter the size :  ");
   scanf("%d",&n1);
   for(i=0;i<n1;i++)
   {
```

```
        printf("Enter the value : ");
        scanf("%f",&y[i]);
    }
    printf("\n\tEnter the value of x to find  f(x) : ");
    scanf("%f",&X);
    printf("|A\t|B\t|C\t|D\t\t|Dr\t|(Yr/Dr)");

printf("\n==========================================================================
==========");
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
        {
            if(i==j)
                lag[i][j]= X-x[j]; //calculating the  difference
            else
                lag[i][j]= x[i]-x[0+j];
        }
    }

    for(i=0;i<n;i++)
    {
```

```
    for(j=0;j<n;j++)
    {
        prod = prod * lag[i][j]; //calculating the row product
    }
    Dr[i] = prod;
    Ydr[i]=(y[i]/Dr[i]); //calculating yr /dr
    prod = 1;
}
printf("\n");
for(i=0;i<n;i++)
{
    for(j=0;j<n;j++)
    {
        printf("%.2f\t",lag[i][j]); //printing the table values
    }
    printf("\t%.2f\t %.2f",Dr[i],Ydr[i]); // printing d r  and yr/dr
    printf("\n");
}
sum=0;
for(i=0;i<n;i++)
{
```

```
    sum=sum+Ydr[i];
  }
  prod=1;
  for(i=0;i<n;i++)
  {
    for(j=0;j<n;j++)
    {
      if(i==j)
      {
        prod =  prod * lag[i][j]; // calculating the answer
      }
    }
  }

printf("\n=============================================================
============");
  printf("\n\n\t The value of f(%.2f) is  ====  %f ",X,(prod*sum)); //printing the answer

printf("\n=============================================================
============");
  return 0;
}
```

OUTPUT :

```
ranges_interpolation }op\cc6 lab> cd "c:\Users\user\

         Enter the size :  4
Enter the value : 5
Enter the value : 6
Enter the value : 9
Enter the value : 11

         Enter the size :  4
Enter the value : 12
Enter the value : 13
Enter the value : 14
Enter the value : 16

        Enter the value of x to find  f(x) : 10
|A      |B      |C      |D              |Dr      |(Yr/Dr)
================================================================================
5.00    -1.00   -4.00   -6.00           -120.00  -0.10
1.00    4.00    -3.00   -5.00           60.00    0.22
4.00    3.00    1.00    -2.00           -24.00   -0.58
6.00    5.00    2.00    -1.00           -60.00   -0.27

================================================================================

         The value of f(10.00) is  ====  14.666667
================================================================================
PS C:\Users\user\Desktop\cc6 lab>
```

PROBLEM STATEMENT

7. Write a program to find out to compute the real roots of the following nonlinear equation by newton Raphson method: correct up to 3 significant digits.

f(x)=x3- 8x - 4

PROGRAM CODE:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
#define f(x) ((x) * (x) * (x)  - 8 * (x) - 4)
#define f1(x) (3 * (x) * (x) - 8 )

int main()
{
   double x,fx,fx1,res,dif, error_tolerance;
   int i = 1;
   printf("\n Enter the value of x :  ");
   scanf("%lf",&x);
   printf("\n please enter error tolerance: ");
   scanf("%lf",&error_tolerance);
   printf("\n-----------------------------------------------------------------\n");
   printf("N\tXn\t\tf(Xn)\t\tf'(Xn)\t\tXn+1");
   printf("\n-----------------------------------------------------------------\n");


   do
   {
      fx = f(x);
      fx1 = f1(x);
      if(fx1 == 0)
      {
         printf("\n please input another value of x : ");
```

```
            break;
        }
        x -= fx/fx1;
        printf("\n%d\t%lf\t%lf\t%lf\t%lf\t",i,x,fx,fx1,x);
        i++;
    }while(fabs(fx)  >= error_tolerance);

    printf("\n----------------------------------------------------------------\n");

    printf("\n\tThe approximate root of the equation is  :  %lf \n",x);

return 0;
}
```

OUTPUT:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\user> cd "c:\Users\user\Downloads\" ; if ($?) { gcc newton_raphson2.c -o newton_raphson2 } ; if ($?) { .\newton_raphson2 }

 Enter the value of x :  2

 please enter error tolerance: 0.001

-----------------------------------------------------------------
N       Xn              f(Xn)           f'(Xn)          Xn+1
-----------------------------------------------------------------

1       5.000000        -12.000000      4.000000        5.000000
2       3.791045        81.000000       67.000000       3.791045
3       3.217045        20.156615       35.116061       3.217045
4       3.062670        3.558054        23.048134       3.062670
5       3.051432        0.226323        20.139842       3.051432
6       3.051374        0.001159        19.933719       3.051374
7       3.051374        0.000000        19.932654       3.051374
-----------------------------------------------------------------

        The approximate root of the equation is  :  3.051374
PS C:\Users\user\Downloads>
```

PROBLEM STATEMENT

8. Write a program to solve for x3+ 2x - 2 = 0 up to three places of decimal, using RegulaFalsi Method.

PROGRAM CODE:

```
#include<stdio.h>

#include<conio.h>

#include<math.h>

#define f(x) ((x) * (x)* (x) + 2 * (x) - 2)


int main()

{

    float a,b,c,tolerance;

    int i = 1;

    do{

    printf("\nEnter the value of a: ");


    scanf("%f",&a); //entering the value of  a


    printf("\nEnter the value of b: ");


    scanf("%f",&b); // entering the value of b
```

```
    printf("\nEnter the tolerance value : ");

    scanf("%f",&tolerance); //entering the tolerance value

   }while( f(a) * f(b) > 0 );

printf("\n\t--------------------------------------------------------------------------------------------------------
------------------\n");

   printf("\n  N   Xn      An      Bn      f(An)       f(Bn)       X(n+1)      f(X(n+1))");

printf("\n\t--------------------------------------------------------------------------------------------------------
------------------\n");

   a = c;

   /*calculating the  root */

   do{

      printf("\n\t%d\t%f\t%f\t%f\t%f\t%f\t%f\t%f",i,c,a,b,f(a),f(b),c,f(c));
```

```
        c = a - ((a - b) / ( f(a) - f(b) )) * f(a);


        if( f(a) * f(c) < 0)
        b = c;


        else
        a = c;


        i++;
    }while( fabs( f(c) ) > tolerance);



printf("\n\t--------------------------------------------------------------------------------------------------------
----------\n");


    printf("\n\nThe approximate root of the equation is :  %g",c); //printing the root


return 0;
}
```

OUTPUT :

```
PS C:\Users\user\Desktop\cc6 lab> cd "c:\Users\user\Desktop\cc6 lab\" ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; i
rFile }

Enter the value of a:
0

Enter the value of b: 1

Enter the tolerance value : 0.0001

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

        N       Xn              An              Bn              f(An)           f(Bn)           X(n+1)          f(X(n+1))
        ------------------------------------------------------------------------------------------------------------------------

        1       0.000000        0.000000        1.000000        -2.000000       1.000000        0.000000        -2.000000
        2       0.666667        0.666667        1.000000        -0.370370       1.000000        0.666667        -0.370370
        3       0.756757        0.756757        1.000000        -0.053106       1.000000        0.756757        -0.053106
        4       0.769023        0.769023        1.000000        -0.007156       1.000000        0.769023        -0.007156
        5       0.770664        0.770664        1.000000        -0.000956       1.000000        0.770664        -0.000956
        6       0.770883        0.770883        1.000000        -0.000128       1.000000        0.770883        -0.000128
        ------------------------------------------------------------------------------------------------------------------------


The approximate root of the equation is :  0.770912
PS C:\Users\user\Desktop\cc6 lab>
```

PROBLEM STATEMENT

9. Write a program to compute the real roots of the following nonlinear equation by Secant Method: x3 +2x -2=0, correct up to 3 significant figures.

PROGRAM CODE:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>

#define f(x) ((x) * (x) * (x) + 2 * (x) - 2)// defining the function in macro form


int main()
{
   double x,fx,fx1,xn,error_tolerance;
   int i = 1;
   printf("\n Enter the value of X :  ");
   scanf("%lf",&x);// taking the xn as input
   printf("\n Enter the value of Xn-1 :  ");
   scanf("%lf",&xn);// taking the value of xn-1 as the input
   printf("\n please enter error tolerance: ");
```

```
scanf("%lf",&error_tolerance);//taking the tolerance value

printf("\n----------------------------------------------------------------\n");
printf ("N\tXn-1\t\tXn\t\tXn+1");
printf("\n----------------------------------------------------------------\n");

do
{
   fx = f(x); // calculating the f(x) by calling the macro function
   fx1 = ( ( f(x) - f(xn) ) / ( x - (xn) ) ); //calculating the fx1 by calling macro
   xn = x;     // updating Xn-1
   x -= fx/fx1; //computing the new value of x i.e Xn+1

    printf ("\n%d\t%lf\t%lf\t%lf\t",i, xn, x,x); //printing the value in  row wise

   i++;       // for printing the number of iteration
}while(fabs(fx)  >= error_tolerance); // the loop will run till the   absolute value of the function is greater than the tolerance the value

printf("\n----------------------------------------------------------------\n");
```

printf("\n\tThe approximate root of the equation is : %lf \n",x);// printing the approximate value of root

return 0;

}

OUTPUT:

```
PS C:\Users\user\Desktop\cc6 lab> cd "c:\Users\user\Desktop\cc6 lab\" ;

 Enter the value of X :
1

 Enter the value of Xn-1 :  0

 please enter error tolerance: 0.0001

-----------------------------------------------------------------
N       Xn-1            Xn              Xn+1
-----------------------------------------------------------------

1       1.000000        0.666667        0.666667
2       0.666667        0.756757        0.756757
3       0.756757        0.771837        0.771837
4       0.771837        0.770909        0.770909
5       0.770909        0.770917        0.770917
-----------------------------------------------------------------

        The approximate root of the equation is  :  0.770917
PS C:\Users\user\Desktop\cc6 lab>
```

PROBLEM STATEMENT

10. Solve the following function using both Trapezoidal and Simpson's 1/3rd Rule using a menu driven program. Use 10 steps in each case. Find the relative percentage error in each case with respect to the ideal value.(consider pi as)

$$\int_0^1 \frac{dx}{1+x^2}$$

PROGRAM CODE:

```
#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

#include<math.h>

#define pi 0.78539816339


#define f(x)  (1 / (1 + (x * x)))

int main()

{

   double i,a,b,h,s = 0.0,g,m,f=0.0,rel;

   double t = 0.0,ig,k;

   int n,choice;


   printf("\n\tEnter the value  of  a  : ");
```

```
scanf("%lf",&a);

printf("\n\tEnter the value  of  b  : ");

scanf("%lf",&b);

printf("\n\tEnter the step size  : ");

scanf("%d",&n);

printf("\n\t*************MENU***************\n");
printf("\n\t1. To Perform Trapezoidal method : \n");
printf("\n\t2. To Perform Simpson's 1/3 rd  method : \n");
printf("\n\t3. To Exit  : \n");
printf("\n\t Enter your choice ..... :  ");
scanf("%d",&choice);
switch (choice)
{

   case 1:
```

```
    h = (b - a) / n;
    g = f(a);
    a = a + h;

    k = 1;
    do
    {
       f = f + f(a);

       a =  a + h;
       k++;

    }while(k!=n);
    i = h/2 *(g + f(b) + 2*f);
    printf("\n-------------------------------------------------");
    printf("\n\tThe integrated value is  : %lf",i);
    rel = ((pi) - i)/(pi);
    printf("\n-------------------------------------------------");

    printf("\n\tThe relative percentage error is  : %lf\n\n",rel);
break;
```

```
case 2 :
    m = (b - a) / n;
    k = a + m;
    for(i=1;i<=(n-1);i=i+2)
    {
        s = s +f(k);
        k =k+(2*m);

    }

    k = a+(2*m);
    for(i=2;i<=(n-2);i=i+2)
    {
        t = t + f(k);
        k = k + (2 * m);

    }

    ig = (m/3) * (f(a) + f(b) + (4 * s) + (2 * t));

    printf("\n-------------------------------------------------");
```

```
        printf("\n\tThe integrated value is  : %lf",ig);


        rel = ((pi) - ig)/(pi);
        printf("\n-------------------------------------------------");
        printf("\n\tThe relative percentage error is  : %lf\n\n",rel);
      break;
      case 3:
        exit(1);
        break;
      default :
        printf("\nInvalid option .....................! ");
        break;
   }


return 0;
}
```

OUTPUT:

```
      Enter the value  of  a  : 0

      Enter the value  of  b  : 1

      Enter the step size  : 10

      **************MENU****************

      1. To Perform Trapezoidal method :


      3. To Exit  :

       Enter your choice ................................ :  1

-------------------------------------------------
      The integrated value is  : 0.784981
-------------------------------------------------
      The relative percentage error is  : 0.000531
```

```
      Enter the value  of  a  : 0

      Enter the value  of  b  : 1

      Enter the step size  : 10

      **************MENU****************

      1. To Perform Trapezoidal method :

      2. To Perform Simpson's 1/3 rd  method :

      3. To Exit  :

       Enter your choice ................................ :  2

-------------------------------------------------
      The integrated value is  : 0.785398
-------------------------------------------------
      The relative percentage error is  : 0.000000
```

PROBLEM STATEMENT

11. The weight of a calf taken at weekly intervals is given below. Fit a straight line using the best squares method and calculate the average rate of growth per week.

Age(x)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Weight(y):	**52.5**	**58.7**	**65.0**	**70.2**	**75.4**	**81.1**	**87.2**	**95.5**	**102.2**	**108.4**

Write a C program to implement the above program. Mention the equation of the line.

PROGRAM CODE:

```
#include<stdio.h>


int main()
{
   int n,i;
   float x[20],y[20],sum=0,sum1=0,sum2=0,X,Y,g,h,k,l,c;
   printf("\n\t Enter the size of x : ");
   scanf("%d",&n);
   i=0;
   while(i<n)
   {
      printf("Enter the data : ");
      scanf("%f",&x[i]);
      i++;
   }
```

```
printf("\n\tEnter the size of y : ");
scanf("%d",&n);
i=0;
while(i<n)
{
   printf("Enter the data : ");
   scanf("%f",&y[i]);
   i++;
}
printf("\n\tXi\tYi\t(Xi - X')\t(Yi - Y')\t(Xi - X')(Yi - Y')\t(Xi - X')2");
for(i=0;i<n;i++)
{
   sum +=x[i];
}
X = sum/n;
sum=0;
for(i=0;i<n;i++)
{
   sum +=y[i];
}
Y = sum/n;
```

```
    for(i=0;i<n;i++)
    {
        g = x[i] - X;
        h = y[i] - Y;
        k=(g*h);
        sum1 += k;
        l=(g*g);
        sum2 += l;
        printf("\n\t%.2f\t%.2f\t%.2f\t\t%.2f\t\t%.2f\t\t\t%.2f ",x[i],y[i],g,h,k,l);
    }
    printf("\n\n\tThe value of m = %.4f ",sum1/sum2);
    c= Y - ((sum1/sum2) * X);
    printf("\n\n\tThe value of c = %.4f ",c);
    printf("\n\t The average rate of growth per week %.4f",sum1/sum2);
    printf("\n\n\t The equation of the line is  y = %.4fx + %.4f",sum1/sum2,c);

    return 0;
}
```

OUTPUT:

```
        Enter the size of x : 10
Enter the data : 1
Enter the data : 2
Enter the data : 3
Enter the data : 4
Enter the data : 5
Enter the data : 6
Enter the data : 7
Enter the data : 8
Enter the data : 9
Enter the data : 10

        Enter the size of y : 10
Enter the data : 52.5
Enter the data : 58.7
Enter the data : 65.0
Enter the data : 70.2
Enter the data : 75.4
Enter the data : 81.1
Enter the data : 87.2
Enter the data : 95.5
Enter the data : 102.2
Enter the data : 108.4
```

```
        Xi      Yi      (Xi - X')       (Yi - Y')       (Xi - X')(Yi - Y')      (Xi - X')2
        1.00    52.50   -4.50           -27.12          122.04                  20.25
        2.00    58.70   -3.50           -20.92          73.22                   12.25
        3.00    65.00   -2.50           -14.62          36.55                   6.25
        4.00    70.20   -1.50           -9.42           14.13                   2.25
        5.00    75.40   -0.50           -4.22           2.11                    0.25
        6.00    81.10   0.50            1.48            0.74                    0.25
        7.00    87.20   1.50            7.58            11.37                   2.25
        8.00    95.50   2.50            15.88           39.70                   6.25
        9.00    102.20  3.50            22.58           79.03                   12.25
        10.00   108.40  4.50            28.78           129.51                  20.25

        The value of m = 6.1624

        The value of c = 45.7267
         The average rate of growth per week 6.1624

         The equation of the line is  y = 6.1624x + 45.7267
PS C:\Users\user\Desktop\cc6 lab>
```

PROBLEM STATEMENT

12. Write a C program to find the roots of a system of linear equations by Gauss elimination method. Test your program with the following:

2x1+3x2+x3=9

x1+2x2+3x3=6

3x1+x2+2x3=8

PROGRAM CODE:

```
#include<stdio.h>

int main()

{

   int n,i,j,k,x,y;

   float ratio,sum;

   printf("\n\tEnter the no. of equations :  ");

   scanf("%d",&n);

   float ar[n][n+1],value[n];

   printf("\n\t Enter the coefficients in row wise form including right hand side : ");

   printf("\n According to the give form input your values from left hand side");

   printf("\nc1x+c2y+c3z..... = b1\nc1x+c2y+c3z.....= b2\nc1x+c2y+c3z.....= b3\n.....");

   for(i=0;i<n;i++)

   {
```

```
    for(j=0;j<n+1;j++)
    {
        printf("\n Enter data : ");
        scanf("%f",&ar[i][j]);
    }
}
for(i=0;i<n;i++)
{
    for(j=0;j<n+1;j++)
    {
        printf("%f ",ar[i][j]);
    }
    printf("\n");
}
// for row manipulation
for(i=0;i<n;i++)
{
    for(j=0;j<n;j++)
    {
        if(j>i)
        {
```

```
            ratio = ar[j][i]/ar[i][i];
            for(k=0;k<n+1;k++)
            {
               ar[j][k] = ar[j][k] - ( ratio * ar[i][k]);
            }
            printf ("\n\tPrinting intermediate matrices\n");
            for(x=0;x<n;x++)
            {
               for(y=0;y<n+1;y++)
               {
                  printf("%f ",ar[x][y]);
               }
               printf("\n");
            }
            printf("\n");
         }
      }
   }
   //for finding value
value[n-1] = ar[n-1][n]/ar[n-1][n-1];
```

```
    for(i=n-2;i>=0;i--)
    {
        sum=0;
        for(j=i+1;j<n;j++)
        {
            sum = sum + (ar[i][j]*value[j]);
        }
        value[i]=(ar[i][n]-sum)/ar[i][i];
    }
    // lastly printing values
    printf("value of x = %f ",value[0]);
    printf("\nvalue of y = %f ",value[1]);
    printf("\nvalue of z = %f ",value[2]);

return 0;
}
```

OUTPUT:

```
PS C:\Users\user\Desktop\cc6 lab> cd "c:\Users\user\Desktop\cc6 lab\" ; if ($?) { gc
on }

       Enter the no. of equations :  3

        Enter the coeffecients in row wise form including right hand side :
 According to the give form input your values from left hand side
c1x+c2y+c3z........ = b1
c1x+c2y+c3z......= b2
c1x+c2y+c3z......= b3
.........
 Enter data : 2

 Enter data : 3

 Enter data : 1

 Enter data : 9

 Enter data : 1

 Enter data : 2

 Enter data : 3

 Enter data : 6

 Enter data : 3

 Enter data : 1

 Enter data : 2

 Enter data : 8
2.000000 3.000000 1.000000 9.000000
1.000000 2.000000 3.000000 6.000000
3.000000 1.000000 2.000000 8.000000
```

```
        Printing intermediate matrices
2.000000 3.000000 1.000000 9.000000
0.000000 0.500000 2.500000 1.500000
3.000000 1.000000 2.000000 8.000000


        Printing intermediate matrices
2.000000 3.000000 1.000000 9.000000
0.000000 0.500000 2.500000 1.500000
0.000000 -3.500000 0.500000 -5.500000


        Printing intermediate matrices
2.000000 3.000000 1.000000 9.000000
0.000000 0.500000 2.500000 1.500000
0.000000 0.000000 18.000000 5.000000

value of x = 1.944445
value of y = 1.611111
value of z = 0.277778
PS C:\Users\user\Desktop\cc6 lab>
```

PROBLEM STATEMENT

13. Write a C program to solve the set of linear equations by Gauss Seidel method:

x-8y+3z=-4

2x-y+9z=12

8x+2y-2z=8

PROGRAM CODE:

```
/*given equations*/

/*x-8y+3z=-4

2x-y+9z=12

8x+2y-2z=8

*/

/*diagonally arranging*/

/*8x+2y-2z=8

x-8y+3z=-4

2x-y+9z=12

*/



#include<stdio.h>

#include<math.h>

#define tolerance 0.001

#define x(y,z) ((8 - 2* y + 2 *z )/8)
```

```
#define y(z,x) ((-4 - 3 * z  -  x)/(-8))

#define z(x,y) ((12 - 2 * x + y)/9)


int main()

{

   double x=0,y=0,z=0,x1,y1,z1;

   int i=0,flag=0;

   printf("\n----------------------------------------------------------------\n");

   printf("\nsteps\tx\ty\tz\n");

   printf("\n%d\t%.2f\t%.2f\t%.2f",i,x,y,z);

   do

   {

      x1=x(y,z);

      y1=y(z,x1);

      z1=z(x1,y1);

      if(fabs(x1-x)<tolerance && fabs(y1-y)<tolerance && fabs(z1-z)<tolerance )

      {

         printf("\n-------------------------------------------------------\n");

         printf("\n\t x = %.3f ",x1);

         printf("\n\t y = %.3f ",y1);

         printf("\n\t z = %.3f ",z1);
```

```
            flag=1;
        }
        else
        {
            x=x1;
            y=y1;
            z=z1;
            i++;
            printf("\n%d\t%.2f\t%.2f\t%.2f",i,x1,y1,z1);

        }

    } while (flag!=1);
    return 0;
}
```

OUTPUT:

```
------------------------------------------------------------

steps   x       y       z

0       0.00    0.00    0.00
1       1.00    0.63    1.18
2       1.14    1.09    1.20
3       1.03    1.08    1.22
4       1.04    1.09    1.22
5       1.03    1.09    1.22
-----------------------------------------------------

         x = 1.034
         y = 1.088
         z = 1.224
PS C:\Users\user\Desktop\cc6 lab>
```

PROBLEM STATEMENT

14. Write a C program to solve the set of linear equations by using gauss Jacobi method:

x+y+4z=9

8x-3y+2z=20

4x+11y-z=33

PROGRAM CODE:

```
#include<stdio.h>

#include<math.h>

#define tolerance 0.001


#define x(y,z) ((20 + 3* y - 2 *z )/8)

#define y(z,x) ((33 + z - 4 * x)/11)

#define z(x,y) ((9 - x - y)/4)


int main()

{

   double x=0,y=0,z=0,x1,y1,z1;

   int i=0,flag=0;

   printf("\n-------------------------------------------------------------\n");

   printf("\nsteps\tx\ty\tz\n");

   printf("\n%d\t%.2f\t%.2f\t%.2f",i,x,y,z);

   do
```

```
{
   x1=x(y,z);
   y1=y(z,x);
   z1=z(x,y);
   if(fabs(x1-x)<tolerance && fabs(y1-y)<tolerance && fabs(z1-z)<tolerance )
   {
      printf("\n-------------------------------------------------------\n");
      printf("\n\t x = %.3f ",x1);
      printf("\n\t y = %.3f ",y1);
      printf("\n\t z = %.3f ",z1);
      flag=1;
   }
   else
   {
      x=x1;
      y=y1;
      z=z1;
      i++;
      printf("\n%d\t%.2f\t%.2f\t%.2f",i,x1,y1,z1);

   }
```

```
    } while (flag!=1);

    return 0;

}
```

OUTPUT:

```
PS C:\Users\user\Desktop\cc6 lab> cd "c:\Users\user\Desktop\c
rFile }

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

steps   x       y       z

0       0.00    0.00    0.00
1       2.50    3.00    2.25
2       3.06    2.30    0.88
3       3.14    1.97    0.91
4       3.01    1.94    0.97
5       2.98    1.99    1.01
6       2.99    2.01    1.01
7       3.00    2.00    1.00
8       3.00    2.00    1.00
-----------------------------------------------------

         x = 3.000
         y = 2.000
         z = 1.000
PS C:\Users\user\Desktop\cc6 lab>
```

PROBLEM STATEMENT

15. Write a C program to solve the set of linear equations by using Gauss-Jordan method:

3x-5y+6z=11

x+y-z=0

2x-y+4z=12

PROGRAM CODE:

```
#include<stdio.h>
int main()
{
   int n,i,j,k,x,y;
   float ratio,sum;
   printf("\n\tEnter the no. of equations :  ");
   scanf("%d",&n);
   float ar[n][n+1],value[n];
   printf("\n\t Enter the coefficients in row wise form including right hand side : ");
   printf("\n\tAccording to the give form input your values from left hand side");
   printf("\nc1x+c2y+c3z..... = b1\nc1x+c2y+c3z.....= b2\nc1x+c2y+c3z.....= b3\n.....");
   for(i=0;i<n;i++)
   {
      for(j=0;j<n+1;j++)
      {
         printf("Enter data : ");
         scanf("%f",&ar[i][j]);
      }
   }
   for(i=0;i<n;i++)
   {
      for(j=0;j<n+1;j++)
      {
         printf("%f ",ar[i][j]);
```

```
        }
        printf("\n");
    }
    // for row manipulation
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
        {
            if(j!=i)
            {
                ratio = ar[j][i]/ar[i][i];
                for(k=0;k<n+1;k++)
                {
                    ar[j][k] = ar[j][k] - ( ratio * ar[i][k]);
                }
                printf("\n\t Printing intermediate matrices \n");
                for(x=0;x<n;x++)
                {
                    for(y=0;y<n+1;y++)
                    {
                        printf("%f ",ar[x][y]);
                    }
                    printf("\n");
                }
                printf("\n");
            }
        }
    }
    printf("\n The value of x = %f ",ar[0][n]/ar[0][0]);
    printf("\n The value of y = %f ",ar[1][n]/ar[1][1]);
    printf("\n The value of z = %f ",ar[2][n]/ar[2][2]);
    return 0;
}
```

OUTPUT:

```
        Enter the no. of equations :  3

         Enter the coeffecients in row wise form including right hand side :
        According to the give form input your values from left hand side
c1x+c2y+c3z........ = b1
c1x+c2y+c3z......= b2
c1x+c2y+c3z......= b3
..........Enter data : 3
Enter data : -5
Enter data : 6
Enter data : 11
Enter data : 1
Enter data : 1
Enter data : -1
Enter data : 0
Enter data : 2
Enter data : -1
Enter data : 4
Enter data : 12
3.000000 -5.000000 6.000000 11.000000
1.000000 1.000000 -1.000000 0.000000
2.000000 -1.000000 4.000000 12.000000
```

```
        Printing intermediate matrices
3.000000 -5.000000 6.000000 11.000000
-0.000000 2.666667 -3.000000 -3.666667
-0.000000 2.333333 -0.000000 4.666667


        Printing intermediate matrices
3.000000 0.000000 0.375000 4.125000
-0.000000 2.666667 -3.000000 -3.666667
-0.000000 2.333333 -0.000000 4.666667


        Printing intermediate matrices
3.000000 0.000000 0.375000 4.125000
-0.000000 2.666667 -3.000000 -3.666667
-0.000000 -0.000000 2.625000 7.875000


        Printing intermediate matrices
3.000000 0.000000 -0.000000 3.000000
-0.000000 2.666667 -3.000000 -3.666667
-0.000000 -0.000000 2.625000 7.875000


        Printing intermediate matrices
3.000000 0.000000 -0.000000 3.000000
-0.000000 2.666667 0.000000 5.333333
-0.000000 -0.000000 2.625000 7.875000


 The value of x = 1.000000
 The value of y = 2.000000
 The value of z = 3.000000
PS C:\Users\user\Desktop\cc6 lab>
```

PROBLEM STATEMENT

16. Write a C program to compute y for x=0.1 by Euler method, correct up to five significant figures, taking step length=0.02,with initial condition y=1 at x=0 dy/dx=(y-x)/(y+x)

PROGRAM CODE:

```
#include<stdio.h>

#define h 0.01
double f(double x,double y)
{
   double q;
    q = (y-x)/(y+x);
    return q;
}
int main()
{
   double i,yn=1,xn=0,um;
   printf("Enter the upper limit : ");
   scanf("%lf",&um);
   printf("X\t\tY");
   printf("\n");
   printf("----------------------");
   printf("\n%lf\t%lf",xn,yn);
   while(xn < um)
   {

      xn = xn + h;
      yn = yn  + (h *f(xn,yn));
      printf("\n%lf\t%lf",xn,yn);

   }
   printf("\n");
   printf("----------------------");
   printf("\n\tThe Solution  :      %lf ",yn);
   return 0;
```

}

OUTPUT:

```
Enter the upper limit : 0.1
X               Y
----------------------
0.000000        1.000000
0.010000        1.009802
0.020000        1.019414
0.030000        1.028842
0.040000        1.038093
0.050000        1.047174
0.060000        1.056090
0.070000        1.064847
0.080000        1.073450
0.090000        1.081903
0.100000        1.090210
0.110000        1.098377
----------------------
        The Solution  :      1.098377
PS C:\Users\user\Desktop\cc6 lab>
```

PROBLEM STATEMENT

17. Write a C program to compute y for x=0.1 by Euler Modified Method, correct up to five significant figures, taking step length h=0.05

dy/dx = (y + x) with initial condition y=1 at x=0

PROGRAM CODE:

```
#include<stdio.h>
#define f(x,y) ( y + x )

int main()
{
        double ul,h,x=0,y=1,yn,xn=0,yn1;
        printf("\n\t Enter the upper limit  :  ");
        scanf("%lf",&ul);
        printf("\n\t Enter the step size ( h ) :   ");
        scanf("%lf",&h);
        printf("\n\tX\t\tY");
        printf("\n");
        printf("-----------------------------------");
        yn = y + h * f(x,y);
        printf("\n\t%lf\t%lf",x,yn);
        while(xn < ul)
        {
                xn = xn + h;
                yn1 = y + ((h / 2) * ( f(x,y) + f(xn,yn) ));
                yn=yn1;
                printf("\n\t%lf\t%lf",xn,yn);

        }
        printf("\n");
        printf("-----------------------------------");
        printf("\n\tThe Solution  :      %lf ",yn1);
        return 0;
}
```

OUTPUT:

```
        Enter the upper limit  :  0.1

        Enter the step size ( h ) :   0.05

       X               Y
-----------------------------------
       0.000000        1.050000
       0.050000        1.052500
       0.100000        1.053813
-----------------------------------
       The Solution  :      1.053813
-------------------------------
Process exited after 5.19 seconds with return value 0
Press any key to continue . . .
```

PROBLEM STATEMENT

18. Write a C program to compute y for x=0.4 by Runge Kutta Method, correct up to six significant figures, taking step length h=0.1
dy/dx = (x-y) with initial condition y=1 at x=0

PROGRAM CODE:

```
#include<stdio.h>
#define f(x,y)  ( x - y )

int main()
{
   double yo=1,xo=0,ul,h,k1,k2,k3,k4;

   printf("Enter the step size  : ");
   scanf("%lf",&h);
   printf("Enter the upper limit : ");
   scanf("%lf",&ul);
   printf("\n\tX\t\tY");
   printf("\n*************************************************************************");
   printf("\n\t%lf\t%lf",xo,yo);
   while( xo < ul )
   {

      k1= h* f(xo,yo);
      k2= h* f((xo+(h/2)),(yo+(k1/2)));
      k3= h* f((xo+(h/2)),(yo+(k2/2)));
      k4= h* f((xo+h),(yo+k3));
      yo= yo + ((k1+(2*k2)+(2*k3)+k4)/6);
      xo+=h;
      printf("\n\t%lf\t%lf",xo,yo);
   }
   printf("\n*************************************************************************");
   printf("\n\tThe solution  is  : %lf",yo);
   return 0;
}
```

OUTPUT:

```
Enter the step size  : 0.1
Enter the upper limit : 0.4

        X               Y
*************************************************************************
        0.000000        1.000000
        0.100000        0.909675
        0.200000        0.837462
        0.300000        0.781637
        0.400000        0.740641
*************************************************************************
        The solution  is  : 0.740641
PS C:\Users\user\Desktop\cc6 lab>
```