LABORATORY MANUAL

APPLIED NUMERICAL TECHNIQUES AND COMPUTING LAB.

ME-321-F

LIST OF EXPERIMENTS

APPLIED NUMERICAL TECHNIQUES AND COMP.

SR.NO.	NAME OF EXPERIMENTS	DATE	SIGNATURE
1.	Solution of Non-linear equation in single variable using the method of successive bisection.		
2.	Solution of Non-linear equation in single variable using the Regula falsi method		
3.	Solution of Non-linear equation in single variable using the Newton Raphson method.		
4.	Solution of a system of simultaneous algebraic equation using Gauss Elimination method.		
5.	Solution of a system of simultaneous algebraic equation using Gauss Seidal method.		
6.	Numerical solution of an ordinary differential equation using the Euler's method.		
7.	Numerical solution of an ordinary differential equation using the Runge Kutta method.		
8.	Numerical solution of an ordinary differential equation using the Milne predicator corrector method.		
9	Numerical solution of a system of two ordinary differential equation using Numerical intergration.		
10	Numerical solution of an elleptic boundary value problem using the method of Finite Differences.		

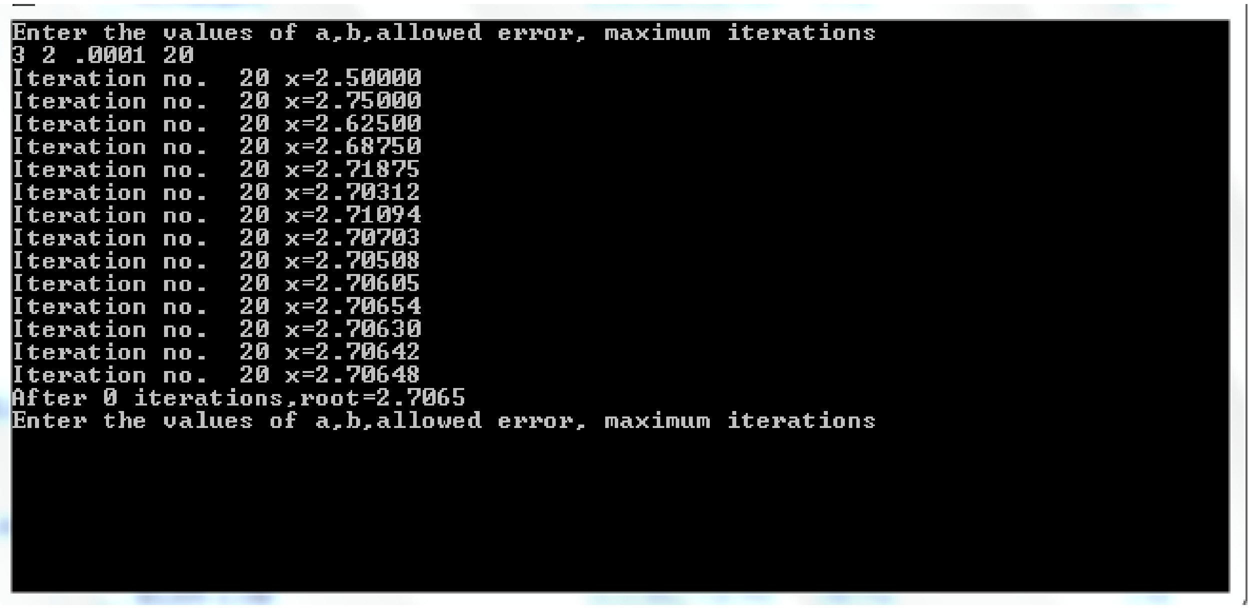
EXPERIMENT NO.1

AIM:-

Solution of Non-linear equation in single variable using the method of successive bisection.

PROGRAM:-

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
float f(float x)
{
return (x*x*x-4*x-9);
}
void bisect(float*x,float a,float b,int *itr)
{
*x= (a+b)/2;
++(itr);
printf("Iteration no. %3d x=%7.5f\n",*itr,*x);
}
main()
{
int itr =0,maxitr;
float x,a,b,aerr,x1;
printf("Enter the values of a,b,""allowed error, maximum iterations\n");

scanf("%f %f %f %d",&a,&b,&aerr,&maxitr);

bisect(&x,a,b,&itr);

do

{

if (f(a)*f(x) < 0)

b=x;

else

a=x;

bisect (&x1,a,b,&itr);

if (fabs(x1-x) <aerr)

{

printf("After %d iterations,root"

"=%6.4f\n",itr,x1);

return 0;

}

x=x1;

} while (itr < maxitr);

printf("Solution does not converge,"

"iterations not sufficient");

return 1;

}
```



```
Enter the values of a,b,allowed error, maximum iterations
3 2 .0001 20
Iteration no.  20 x=2.50000
Iteration no.  20 x=2.75000
Iteration no.  20 x=2.62500
Iteration no.  20 x=2.68750
Iteration no.  20 x=2.71875
Iteration no.  20 x=2.70312
Iteration no.  20 x=2.71094
Iteration no.  20 x=2.70703
Iteration no.  20 x=2.70508
Iteration no.  20 x=2.70605
Iteration no.  20 x=2.70654
Iteration no.  20 x=2.70630
Iteration no.  20 x=2.70642
Iteration no.  20 x=2.70648
After 0 iterations,root=2.7065
Enter the values of a,b,allowed error, maximum iterations
```

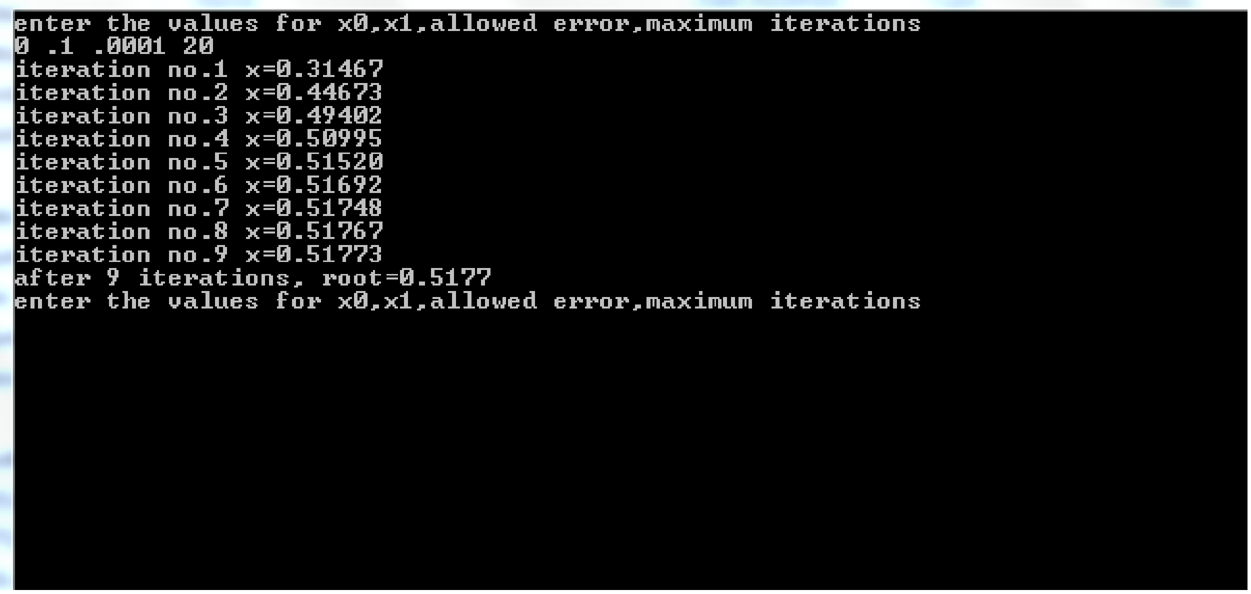
EXPERIMENT NO.2

AIM:-

Solution of Non-linear equation in single variable using the Regula falsi method.

PROGRAM:-

```
#include<stdio.h>
#include<math.h>
float f (float x)
{
return cos(x)-x*exp(x);
}
void regula (float *x, float x0,float x1,float fx0,float fx1,int *itr)
{
*x=x0-((x1-x0)/(fx1-fx0))*fx0;
++(*itr);
printf("iteration no. %3d x=%7.5\n",
*itr,*x);
}
main()
{
int itr=0, maxitr;
float x0,x1,x2,x3,aerr;
printf("Enter the values for x0,x1,"
"allowed error,maximum iterations\n");
scanf("%f%f%f%d",&x0,&x1,&aerr,&maxitr);
regula(&x2,x0,x1,f(x0),f(x1),&itr),
do
{if (f(x0)*f(x2)<0)
x1=x2;
else
x0=x2;
regula(&x3,x0,x1,f(x0),f(x1),&itr);
if (fabs(x3-x2)<aerr)
{
printf("after %d iterations,"
"root=%6.4f\n",itr,x3);
return 0;
}
```



```
enter the values for x0,x1,allowed error,maximum iterations
0 .1 .0001 20
iteration no.1 x=0.31467
iteration no.2 x=0.44673
iteration no.3 x=0.49402
iteration no.4 x=0.50995
iteration no.5 x=0.51520
iteration no.6 x=0.51692
iteration no.7 x=0.51748
iteration no.8 x=0.51767
iteration no.9 x=0.51773
after 9 iterations, root=0.5177
enter the values for x0,x1,allowed error,maximum iterations
```

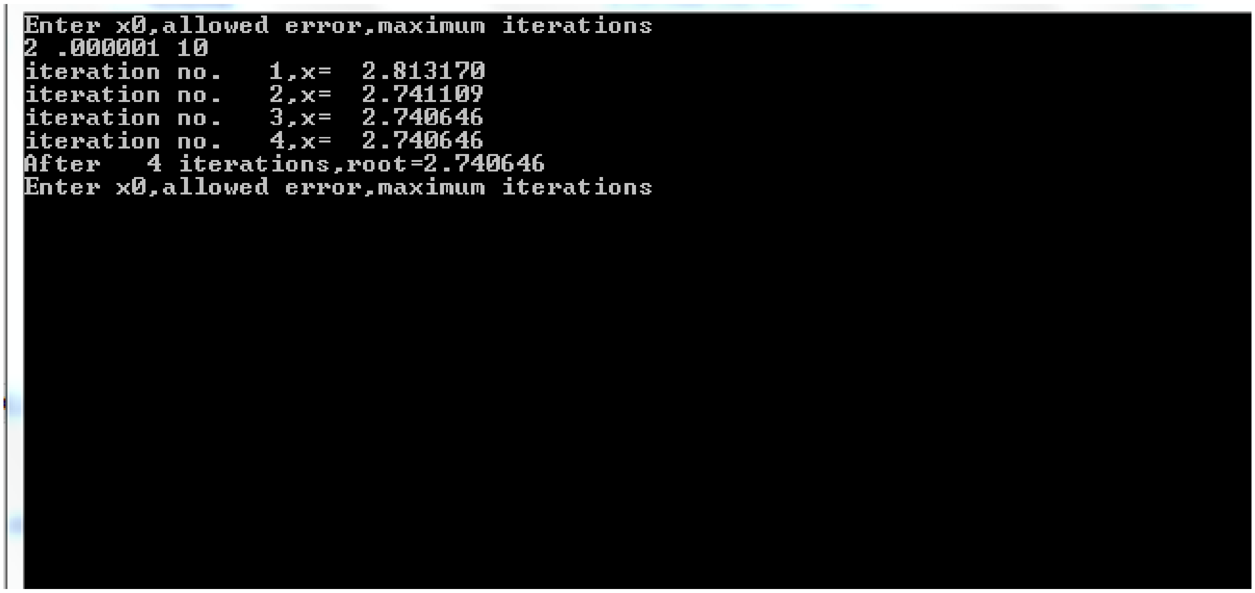
EXPERIMENT NO.3

AIM:-

Solution of Non-linear equation in single variable using the Newton Raphson method.

PROGRAM:-

```
#include<stdio.h>
#include<math.h>
float f(float x)
{
return x*log10(x)-1.2;
}
float df(float x)
{
return log10(x)+0.43429;
}
main()
{
int itr,maxitr;
float h,x0,x1,aerr;
printf("Enter x0,allowed error,"
"maximum iterations\n");
scanf("%f %f %d",&x0,&aerr,&maxitr);
for (itr=1;itr<maxitr;itr++)
{
h=f(x0)/df(x0);
x1=x0-h;
printf("iteration no. %3d,"
"x= %9.6f\n",itr,x1);
if (fabs(h)<aerr)
{
printf ("After %3d iterations,"
"root=%8.6f\n",itr,x1);
return 0;
}
x0=x1;
}
printf ("iterations not sufficient,"
"solution does not converge\n");
return 1;
}
```



```
Enter x0,allowed error,maximum iterations
2 .000001 10
iteration no.   1,x=  2.813170
iteration no.   2,x=  2.741109
iteration no.   3,x=  2.740646
iteration no.   4,x=  2.740646
After   4 iterations,root=2.740646
Enter x0,allowed error,maximum iterations
```

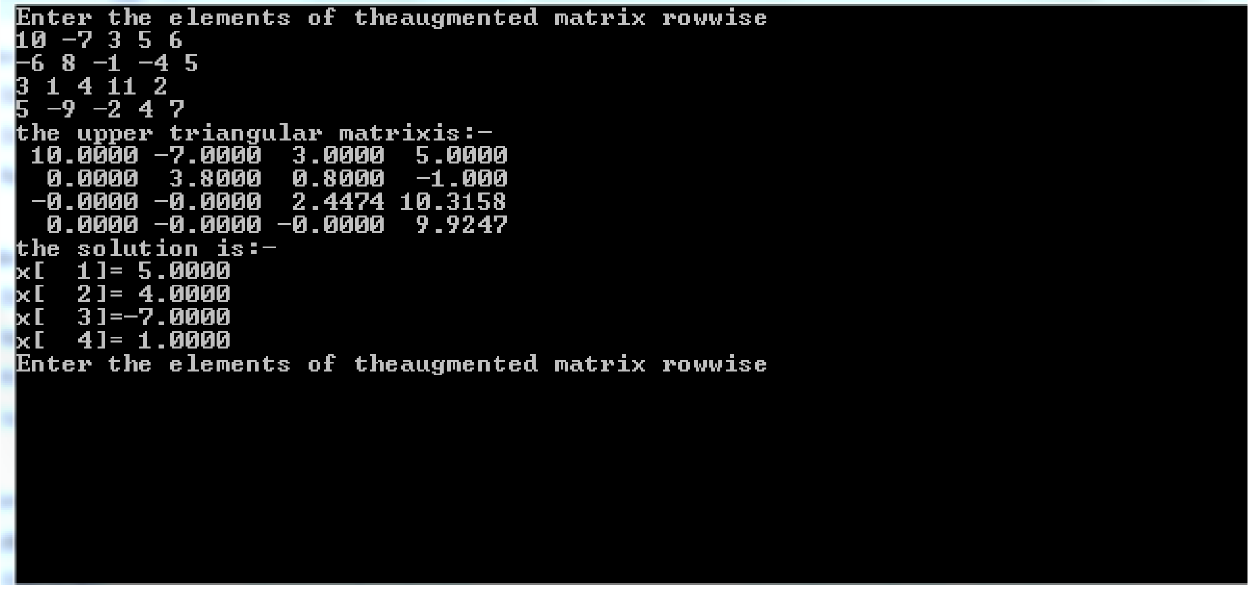
EXPERIMENT NO.4

AIM:-

Solution of a system of simultaneous algebraic equation using Gauss Elimination method.

PROGRAM:-

```
#include<stdio.h>
#define N 4
main()
{
float a[N][N+1],x[N],t,s;
int i,j,k;
printf("Enter the elements of the""augmented matrix rowwise\n");
for (i=0;i<N;i++)
for (j=0;j<n+1;j++)
scanf("%f",&a[i][j]);
for(j=0;j<N-1;j++)
for(i=j+1;i<N;i++)
{
t=a[i][j]/a[j][j];
for (k=0;k<N+1;k++)
a[i][k] -=a[j][k]*t;
}

/* now printing the

upper triangular matrix */

printf("the upper triangular matrix""is:-\n");

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

{for (j=0;j<N;j++)

printf("%8.4f",a[i][j]);

printf("\n");

}

/* now performing back substitution */

for (i=N-1;i>=0;i--)

{

s=0;

for (j=i+1;j<N;j++)

s +=a[i][j]*x[j];

x[i]=(a[i][N]-s)/a[i][i];

}

/* now printing the results */

printf("the solution is:-\n");

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

printf("x[%3d]=%7.4f\n",i+1,x[i]);

}
```



```
Enter the elements of theaugmented matrix rowwise
10 -7 3 5 6
-6 8 -1 -4 5
3 1 4 11 2
5 -9 -2 4 7
the upper triangular matrixis:-
 10.0000 -7.0000  3.0000  5.0000
  0.0000  3.8000  0.8000  -1.000
 -0.0000 -0.0000  2.4474 10.3158
  0.0000 -0.0000 -0.0000  9.9247
the solution is:-
x[  1]= 5.0000
x[  2]= 4.0000
x[  3]=-7.0000
x[  4]= 1.0000
Enter the elements of theaugmented matrix rowwise
```

EXPERIMENT NO.5

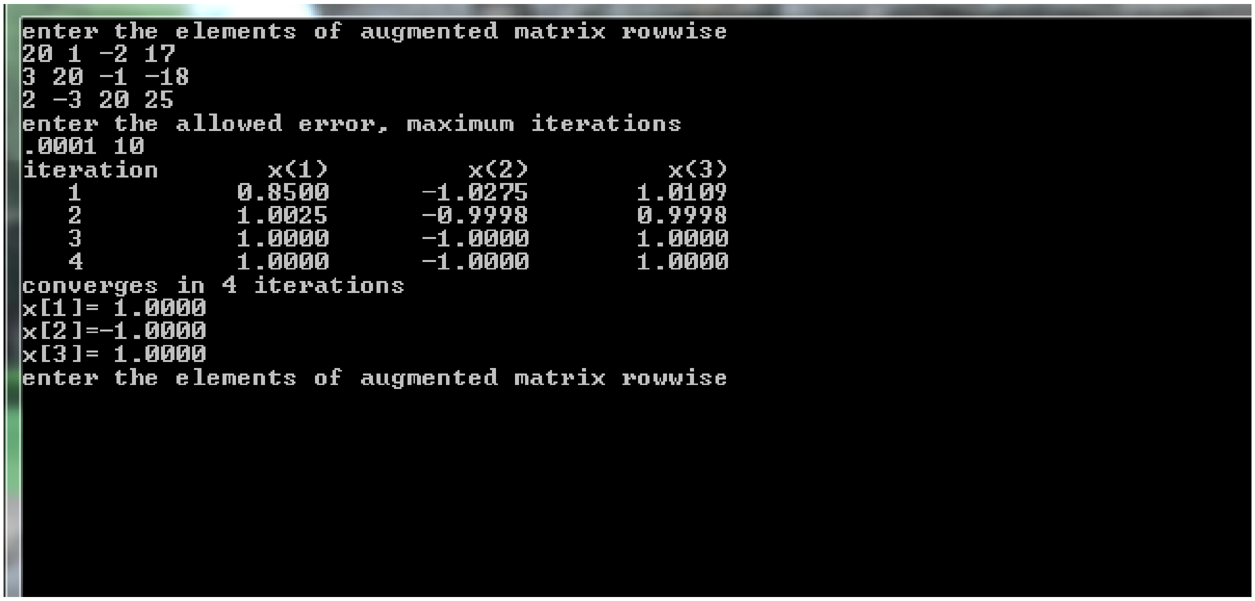
AIM:-

Solution of a system of simultaneous algebraic equation using Gauss Seidal method.

PROGRAM:-

```
#include<stdio.h>

#include<math.h>

#define N 4

main()

{

float a[N][N+1],x[N],aerr,maxerr,

t,s,err;

int i,j,itr,maxitr;

/* first intializing the array x */

for (i=0;i<N;i++) x[i]=0;

printf("Enter the elements of the"

"augmented matrix rowwise\n");

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

for (j=0;j<N;j++)

scanf ("%f",&a[i][j]);
```

```
printf ("enter the allowed error,"
"maximum iterations\n");
scanf("&f %d",&aerr,&maxitr);
printf ("Iteration x[1] x[2]"
"x[3]\n");
for (itr=1;itr<=maxitr;itr++)
{
maxerr=0;
for (i=0;i<N;i++)
{
s=0;
for (j=0;j<N;j++)
if (j!=i) s+=a[i][j]*x[j];
t=(a[i][N]-s)/a[i][i];
err=fabs (x[i]-t);
if (err, maxerr) maxerr=err;
x[i]=t;
}
printf ("%5d",itr);
for (i=20;i<N;i++)
printf("%9.4f",x[i]);
printf("\n");
if (maxerr<aerr)
{
printf("converges in %3d"
"iterations\n",itr);
for(i=0;i<N;i++)
printf("x[%3d]=%7.4f\n",
          i+1,x[i]);
          return 0;
}
}
printf("solution does not converge,"
"iterations not sufficient\n");
return 1;
}
```



```
enter the elements of augmented matrix rowwise
20 1 -2 17
3 20 -1 -18
2 -3 20 25
enter the allowed error, maximum iterations
.0001 10
iteration       x(1)          x(2)          x(3)
   1          0.8500       -1.0275        1.0109
   2          1.0025       -0.9998        0.9998
   3          1.0000       -1.0000        1.0000
   4          1.0000       -1.0000        1.0000
converges in 4 iterations
x[1]= 1.0000
x[2]=-1.0000
x[3]= 1.0000
enter the elements of augmented matrix rowwise
```

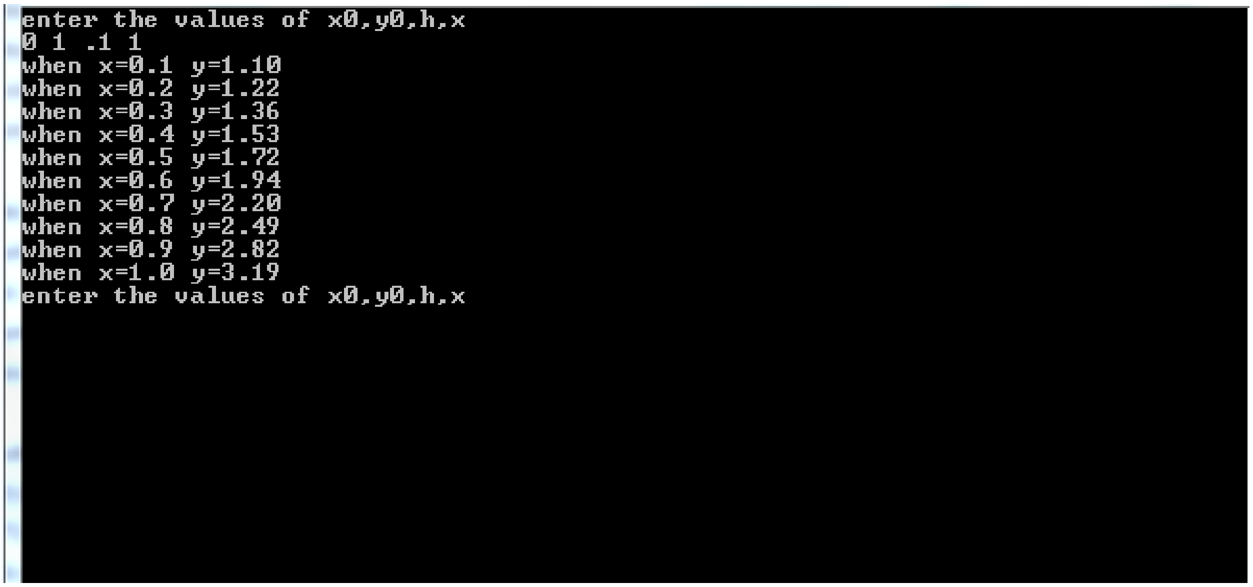
EXPERIMENT NO.6

AIM:-

Numerical solution of an ordinary differential equation using the Euler's method.

PROGRAM:-

```
#include<stdio.h>

float df(float x, float y)

{

return x+y;

}

main()

{

float x0,y0,h,x,x1,y1;

puts("enter the values of x0,y0,h,x");

scanf("%f %f %f %f",&x0,&y0,&h,&x);

x1=x0;y1=y0;

while(1)

{

if(x1>x) return

y1 += h*df(x1,y1);
x1 += h;

printf ("when x=%3.1f"

"y=%4.2f\n",x1,y1);

}

}
```



```
enter the values of x0,y0,h,x
0 1 .1 1
when x=0.1 y=1.10
when x=0.2 y=1.22
when x=0.3 y=1.36
when x=0.4 y=1.53
when x=0.5 y=1.72
when x=0.6 y=1.94
when x=0.7 y=2.20
when x=0.8 y=2.49
when x=0.9 y=2.82
when x=1.0 y=3.19
enter the values of x0,y0,h,x
```

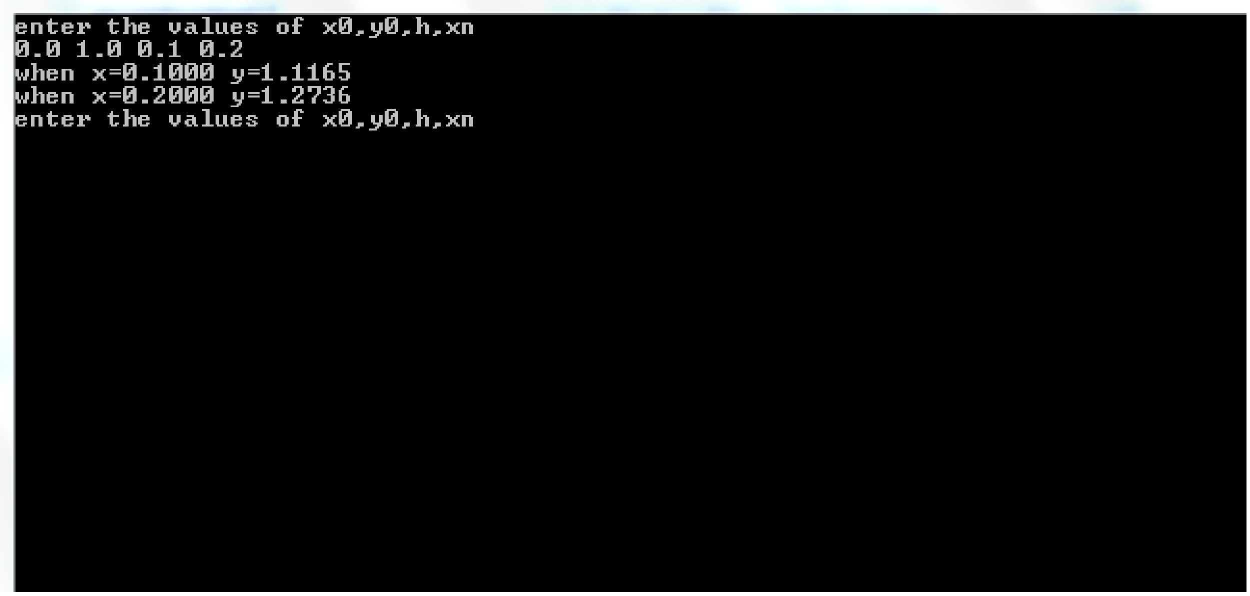
EXPERIMENT NO.7

AIM:-

Numerical solution of an ordinary differential equation using the Runge Kutta method.

PROGRAM:-

```
#include <stdio.h>
float f(float x,float y)
{
return x+y*y;
}
main()
{ float x0,y0,h,xn,x,y,k1,k2,k3,k4,k;
printf ("enter the values of x0,y0,""h,xn\n");
scanf("%f%f%f%f",&x0,&y0,&h,xn);
x=x0;y=y0;
while(1)
{
if (x==xn) break;
k1 =h*f(x,y);
k2 =h*f(x+h/2,y+k1/2);
k3 =h*f(x+h/2,y+k2/2);
k4 =h*f(x+h,y=k3);
k =(k1+(k2+k3)*2+k4)/6;
x += h; y+=k;
printf ("when x=%8.4f""y =%8.4f\n",x,y);
}
}
```



```
enter the values of x0,y0,h,xn
0.0 1.0 0.1 0.2
when x=0.1000 y=1.1165
when x=0.2000 y=1.2736
enter the values of x0,y0,h,xn
```

EXPERIMENT NO.8

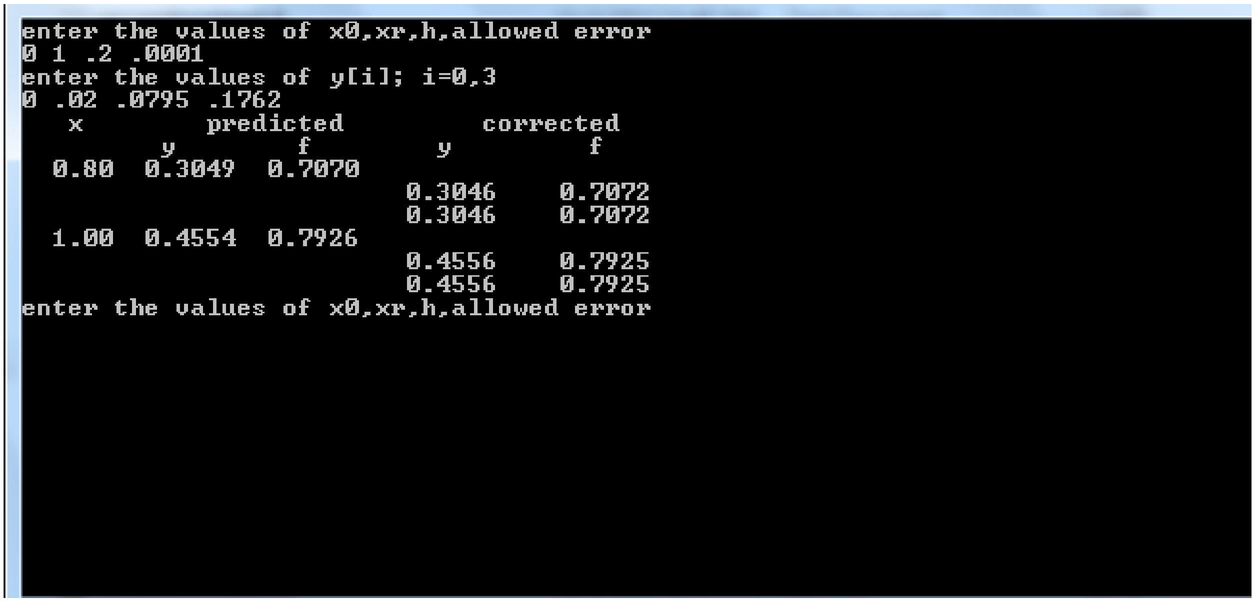
AIM:-

Numerical solution of an ordinary differential equation using the Milne predicator corrector method.

PROGRAM:-

```
#include<stdio.h>

#include<math.h>

float x[5],y[5],h;

float f(int i)

{

 return x[i]-y[i]*y[i];

}

void correct()

{

y[4]=y[2]+(h/3)*(f(2)+4*f(3)+f(4));

printf("%23s %8.4f %8.4f \n", "",y[4],f(4));

}
```

```
main()

{

float xr,aerr,yc;

int i;

puts("enter the value of x0,xr,h,""allowed error");

scanf("%f %f %f %f",&x[0],&xr,&h,&aerr);

puts("enter the value of y[i],i=0,3");

for (i=0;i<=3;i++) scanf("%f",&y[i]);

for (i=1;i<=3;i++) x[i]= x[0]+i*h;

puts("  x predicted""corrected");

puts("  y f""y f");

while (1)

{

 if(x[3] = xr) return;

 x[4]=x[3]+h;

 y[4]=y[0]+

  (4*h/3)*(2*(f(1)+f(3))-f(2));

printf("%6.2f %8.4f\n", x[4],y[4],f(4));

correct();

while(1)

 {

   yc= y[4];

   correct();
  if (fabs(yc-y[4]) <=aerr) break;
 }
 for (i=0;i<=3;i++)
 {
  x[i] =x[i+1];
  y[i] =y[i+1];
  }
 }
}
```



```
enter the values of x0,xr,h,allowed error
0 1 .2 .0001
enter the values of y[i]; i=0,3
0 .02 .0795 .1762
   x         predicted          corrected
          y         f          y          f
  0.80  0.3049  0.7070
                         0.3046    0.7072
                         0.3046    0.7072
  1.00  0.4554  0.7926
                         0.4556    0.7925
                         0.4556    0.7925
enter the values of x0,xr,h,allowed error
```