Numerical Analysis
Practical File

Session: 2022-23

Name: Kartavya Gupta

Course: B.Sc.(Hons) Electronics

Semester: V

Roll No. 1620016

INDEX

Experiment no. 1

Aim:

Program to implement Bisection Method

Algorithm:

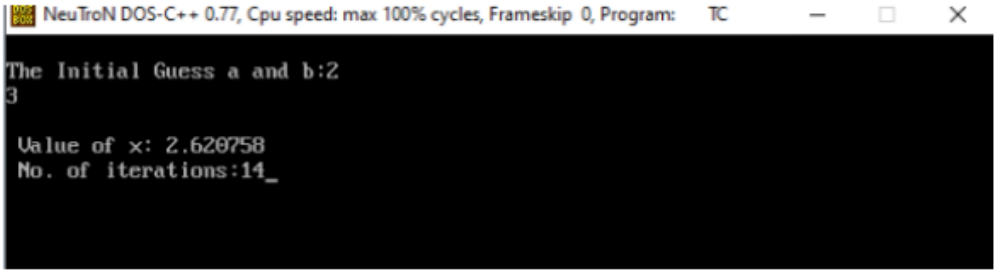
1. Start
2. Choose ‘a’ and ‘b’ initial guesses such that f(a)*f(b)<0 or they have opposite signs.
3. If f(a)*f(b)<0 then proceed further otherwise choose the value of ‘a’ and ‘b’ again.
4. Find:
5. $x_m = \frac{a+b}{2}$
6. If $x_m = 0$ then x_m is the root of the equation otherwise proceed further
7. If f(a)*f(x_m)<0 then a=a ; b=x_m
8. If f(a)*f(x_m)>0 then a=x_m ; b=b
9. Check if |a-b| ≤ 0.0001 (acceptable error), if it is true then end otherwise go to step 3.

Input:

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

float f(float x)
{
float y=pow(x,3)-18.0;
return y;
}

void main()
{
clrscr();
float a,b,xm;
getthenumber:printf("\nThe Initial Guess a and b:");
scanf("%f%f",&a,&b);

if((f(a)*f(b))>0)
{
printf("The interval is not valid,please enter again");
gotogetthenumber:
}

xm=(a*(f(b)-b*f(b))/(f(b)-f(a));
int count =0;

//using while loop
while ((b-a)>0.0001)
{
```

```
if ((f(a)*f(xm))<0)
{
b=xm;
}
else if ((f(a)*f(xm))>0)
{
a=xm;
}

else if((f(a)*f(xm))==0){

break;
}

xm=((a+b)/2);
count++;
}

printf("\n Value of x: %f",xm);
printf("\n No. of iterations:%d",count);
getch();
}
```

Output:



Experiment no. 2

Aim:

Program to implement Regula falsi method

Algorithm:

1. Choose ‘a’ and ‘b’ initial guesses such that f(a)*f(b)<0 or they have opposite signs.
2. If f(a)*f(b)<0 then proceed further otherwise choose the value of ‘a’ and ‘b’ again.
3. Find $x_n = \frac{af(b)-bf(a)}{f(b)-f(a)}$.
4. If f(a)*f(x_m)<0 then a=a; b=x_m
5. If f(a)*f(x_m)>0 then a=x_m ; b=b
6. Check if |a-b| ≤ 0.0001 (acceptable error), if it is true then end otherwise go to step 3.

Input:

```
//Program to implement Regula falsi method
#include<stdio.h>
#include<math.h>
#define    f(x)   x*exp(x) - 1
int main()
{
     float a, b, xn, f0, f1, f2, e;
     int itr = 1;
     up:
     printf("\nEnter the values of a and b:\n");
     scanf("%f%f", &a, &b);
     printf("Enter  error:\n");
     scanf("%f", &e);
    // Calculating  Values
     f0 = f(a);
     f1 = f(b);
     /* Checking of valued. */
     if( f0*f1 > 0.0)
     {
            printf("Incorrect Initial Guesses.\n");
            goto up;
     }
     /* using regular falsi method*/
```

```
    printf("\nitr\t\ta\t\tb\t\txn\t\tf(xn)\n");
    do
    {
        xn = (a*f1-b*f0)/(f1-f0);
        f2 = f(xn);
        printf("%d\t\t%f\t%f\t%f\t%f\n",itr, a, b, xn, f2);
        if(f0*f2 < 0)
        {
            b = xn;
            f1 = f2;
        }
        else
        {
            a = xn;
            f0 = f2;
        }
        itr = itr + 1;
    }while(fabs(f2)>e);
    printf("\nRoot is: %f", xn);
    return 0;
}
```

Output:

```
C:\TURBOC3\BIN>TC

Enter the values of a and b:
0
1
Enter  error:
0.1

itr             a               b               xn              f(xn)
1               0.000000        1.000000        0.367879        -0.468536
2               0.367879        1.000000        0.503314        -0.167420
3               0.503314        1.000000        0.547412        -0.053649

Root is: 0.547412_
```

Experiment no. 3

Aim:

Program to implement Newton Raphson Method.

Algorithm:

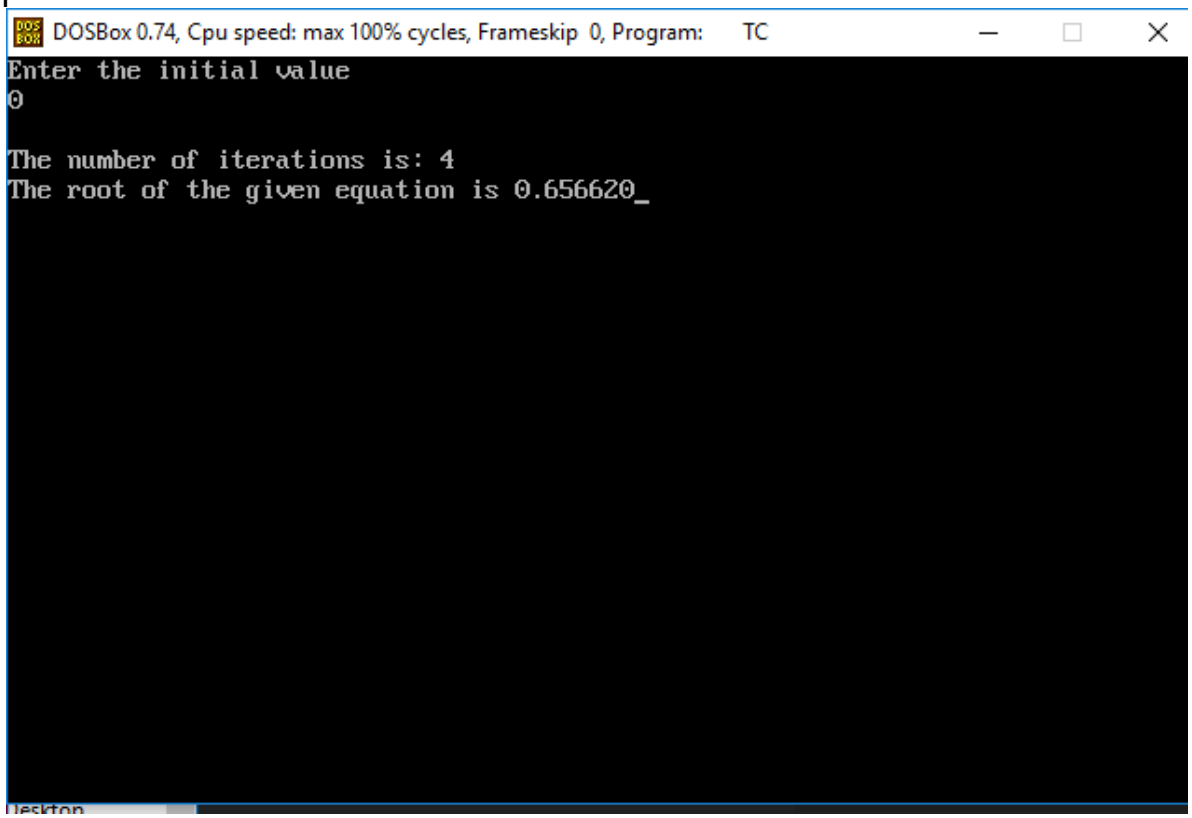
1. Define the function and its derivative
2. Define error and maximum number of iterations
3. Read initial guess x0,
4. If f(x0) =0 then x0 is the root of the equation , exit
5. If f'(x0) =0 then x0 is not valid , exit.
6. Find x1=x0-(f(x0)/f'(x0))
7. Find |x1-x0|
8. If |x1-x0|<error then x1 is the root, else
9. x1=x0 and follow all steps from step 4

Input:

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

float func(float x)
{
float y=pow(x,3)-(5*x)+3.0;
return y;
}
float dif(float x){
float z=3*pow(x,2)-5.0;
return z;}
void main()
{
clrscr();
float x[100];
float a,b;
int count=0;
aa:
printf("Enter the initial value \n");
scanf("%f",&x[0]);

if (dif(x[0])==0){
printf("The value of a is incorrect. \n Press any key to continue!!\n");
goto aa;
}

int n=0;

//using do-while loop
do {
n++;
x[n]=x[n-1]-(func(x[n-1])/dif(x[n-1]));
```

```
count++;
if (dif(x[n])==0){
printf("f(x)=0, Thus, the roots does not exist.\n");
break;}
else{ continue;}
} while (fabs(x[n]-x[n-1])>0.00001);
printf("\nThe number of iterations is: %d",count);
printf("\nThe root of the given equation is %f",x[n]);
getch();
}
```

Output:



Experiment no. 4

Aim:

Program to implement Secant Method.

Algorithm:

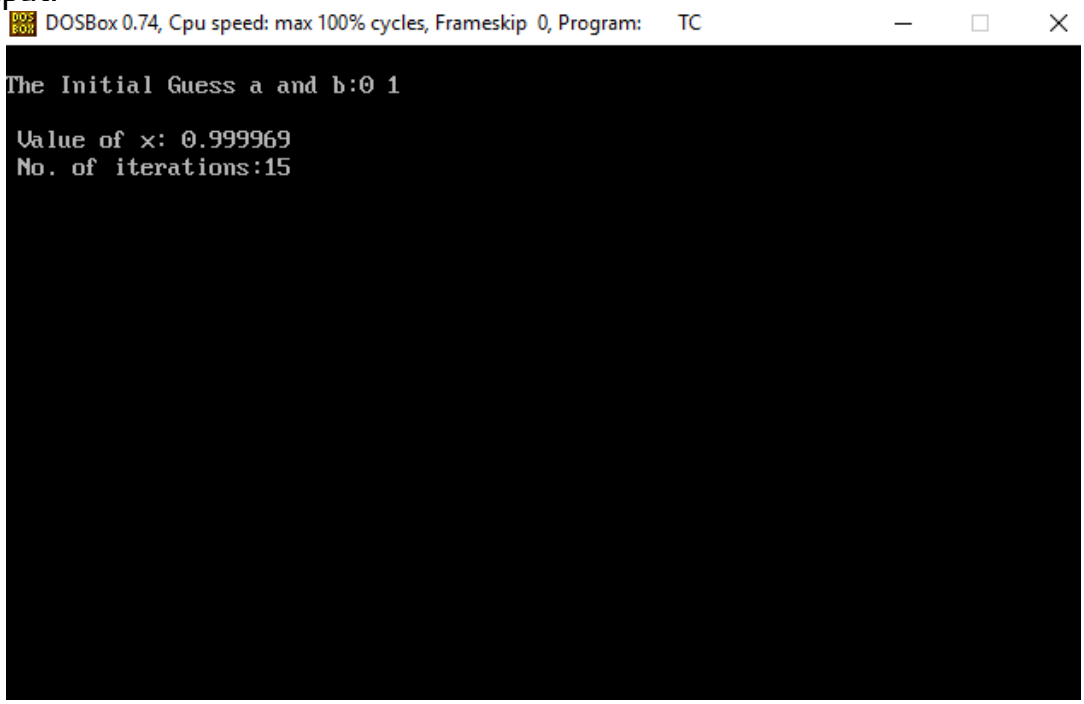
1. Define f(x)
2. Choose ‘a’ and ‘b’ initial guesses such that f(a)*f(b)<0 or they have opposite signs.
3. If f(a)*f(b)<0 then proceed further otherwise choose the value of ‘a’ and ‘b’ again.
4. Find $x_n = \frac{a+b}{2}$.
5. If f(a)*f(x_m)<0 then a=a; b=x_m.
6. If f(a)*f(x_m)>0 then a=x_m ; b=b.

Input:

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

float f(float x)
{
float y=pow(x,3)-18.0;
return y;
}

void main()
{
clrscr();
float a,b,xm;
printf("\nThe Initial Guess a and b:");
scanf("%f%f",&a,&b);

int count =0;

//using while loop
while ((b-a)>0.0001)
{
if ((f(a)*f(xm))<0)
{
b=xm;
}
else if ((f(a)*f(xm))>0)
{
a=xm;
}

else if((f(a)*f(xm))==0){

break;
}
```

```
xm =((a+b)/2);
count++;
}

printf("\n Value of x: %f",xm);
printf("\n No. of iterations:%d",count);
getch();
}
```

Output:



Experiment no. 5

Aim:

Program to solve a problem using LaGrange's interpolation.

Algorithm:

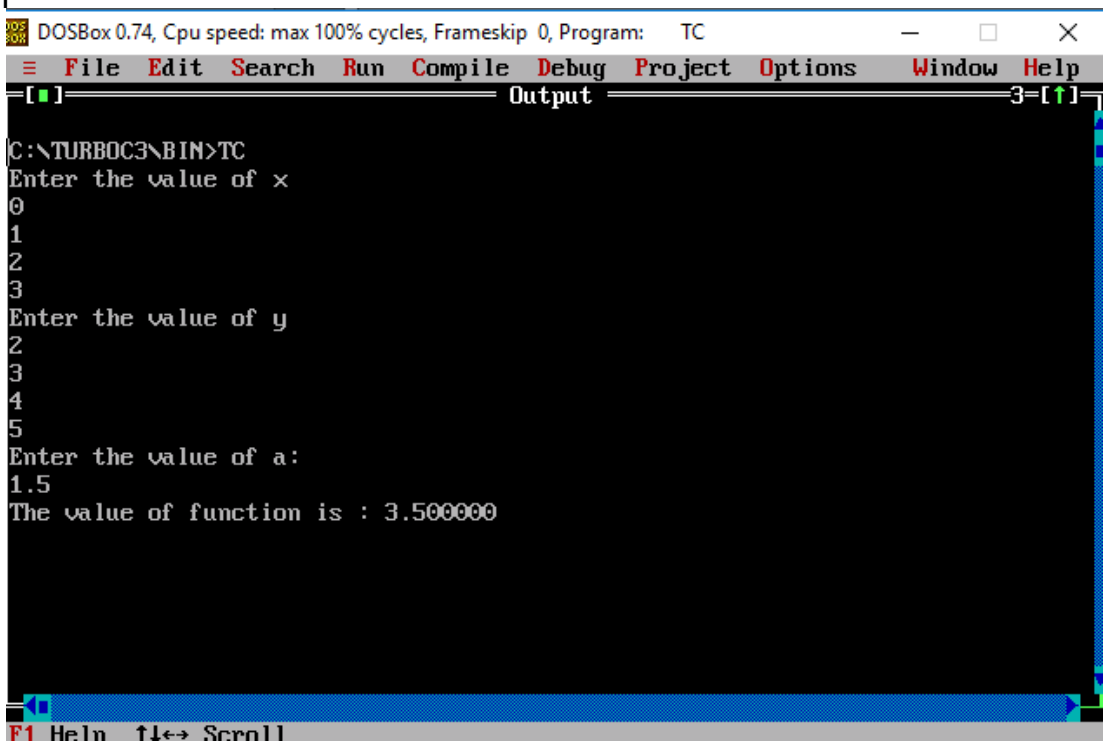
1. Define two array X and Y, int i, a,y
2. Read the values of arrays X and Y where elements of array Y are the corresponding values of Y for X
3. Read the value of a to be calculated
4. Define the formula
5. Result of the formula will be the desired result
6. End

Input:

```
#include<stdio.h>
#include<stdlib.h>
void main()
{
int i,x[4], y[4]; //no of values are 4
double x1,x2,x3,x4,a,y1;
printf("Enter the value of x\n");
for(i=0;i<4;i++)
{
scanf("%d",&x[i]);
}
printf("Enter the value of y\n");
for(i=0;i<4;i++)
{
scanf("%d",&y[i]);
}
printf("Enter the value of a:\n");
scanf("%lf", &a);
x1= ((a-x[1])*(a-x[2])*(a-x[3])*y[0])/((x[0]-x[1])*(x[0]-x[2])*(x[0]-x[3]));
x2=((a-x[0])*(a-x[2])*(a-x[3])*y[1])/((x[1]-x[0])*(x[1]-x[2])*(x[1]-x[3]));
x3=((a-x[0])*(a-x[1])*(a-x[3])*y[2])/((x[2]-x[0])*(x[2]-x[1])*(x[2]-x[3]));
```

```
x4=((a-x[0])*(a-x[1])*(a-x[2])*y[3])/((x[3]-x[0])*(x[3]-x[1])*(x[3]-x[2]));
 y1 =  x1+x2+x3+x4;
printf("The value of function is : %lf",y1);
}
```

Output:



Experiment no. 6

Aim:

To solve the problem using linear regression.

Algorithm:

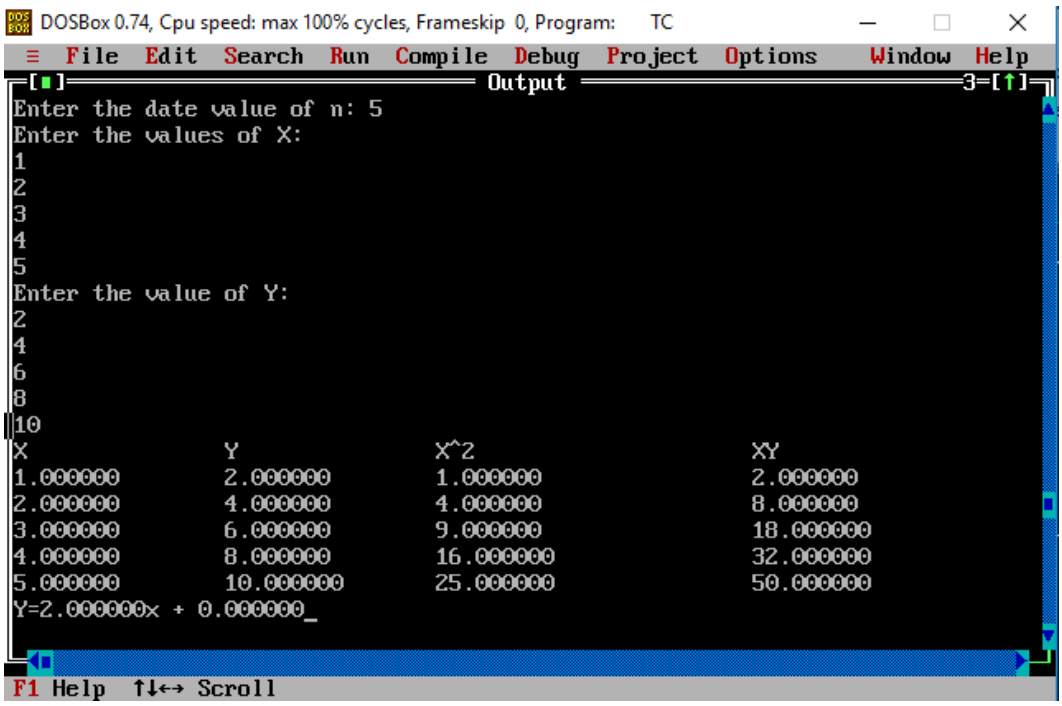
1. Read n (no of data points) from the user and declare a,b.
2. Use for loop to read the values of x and y from the user in x[i],y[i].
3. Initailize sumx1=0, sumx2=0, sumy1=0, sumxy=0
4. Use loop to calculate summation-
   sumx1 = sumx1 + x[i]
   sumx2 = sumx2+ x[i]*x[i]
   sumy1 = sumy1 + y[i]
   sumxy = sumxy + x[i]*y[i]
5. Use a and b-
   a = (n*sumxy-sumx1*sumy1)/(n*sumx2-sumx1*sumx1)
   b= (sumy1*sumx2-sumx1*sumxy)/(n*sumx2-(sumx1*sumx1))
6. Display value of a and b.Read the values of arrays X and Y where elements of array Y are the corresponding values of Y for X

Input:

```
// To solve the solution of linear regression
#include<stdio.h>
#include<stdlib.h>
#include<math.h>

int main()
{
double a0,a1,x[10],y[10];
double sumx=0,sumxy=0,sumx2=0,sumy=0;
int i,n;
printf("Linear Regresion \n");
printf("Enter the date value of n: ");
scanf("%d",&n);

printf("Enter the values of X:\n");
for(i=0;i<n;i++)
{
scanf("%lf",&x[i]);
}

printf("Enter the value of Y:\n");
for(i=0;i<n;i++)
{
scanf("%lf",&y[i]);
}
```

```
for(i=0;i<n;i++)
{
sumx= sumx + x[i];
sumx2 = sumx2 + x[i]*x[i];
sumy= sumy + y[i];
sumxy= sumxy + x[i]*y[i];
}

printf("X\t\tY\t\tX^2\t\t\tXY\n");
for(i=0;i<n;i++)
{
printf("%lf\t%lf\t%lf\t\t%lf\n",x[i],y[i],x[i]*x[i],x[i]*y[i]);
}

a0= ((sumy*sumx2) - (sumx*sumxy))/((n*sumx2)-(sumx*sumx));
a1= ((n*sumxy)-(sumx*sumy))/((n*sumx2)-(sumx*sumx));


printf("Y=%lfx + %lf",a1,a0);

}
```

Output:



Experiment no. 7

Aim:

To solve the problem using implement Trapezoidal rule.

Algorithm:

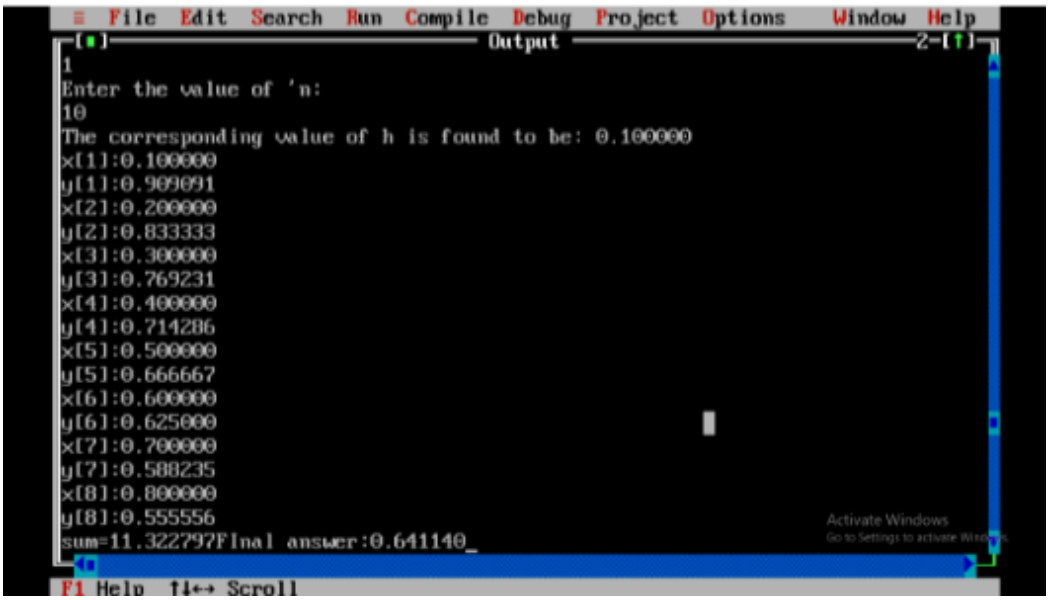
1. Call the function
2. Define h, n,a,b,ans,I,i array X, array Y
3. Define f(x), sum=0
4. Calculate h
5. Using 'for' loop from i=0 to i<n receive the values of array x
6. Using 'for' loop calculate the value of corresponding y as y= f(x[i])
7. Calculate sum of the middle term
8. Calculate integral as I=h[y[0]+y[n]+2*Sum]/2
9. Print the result.

Input:

```
#include <math.h>
#include <stdio.h>
#include <conio.h>
float f(float x){
float y=(1/(1+x));
return y;
}
void main(){
float a, b, h;
float x[10], y[10];
int n,i,j;
printf("Enter the value of 'a':\n");
scanf("%f",&a);
printf("Enter the value of 'b':\n");
scanf("%f",&b);
printf("Enter the value of 'n:\n");
scanf("%d",&n);
h=((b-a)/n);
printf("The corresponding value of h is found to be: %f\n",h);
x[0]=a;
x[n-1]=b;
for(i=1;i<(n-1);i++){
x[i]=x[i-1]+h;
y[i]=f(x[i]);
printf("x[%d]:%f\n",i,x[i]);
printf("y[%d]:%f\n",i,y[i]);
}
float sum,integ;
sum=0.0;
integ=0.0;
for (j=1;j<n-1;j++){
sum+=(2*y[j]);
}
printf("sum=%f",sum);
y[0]=f(x[0]);
```

```
y[n-1]=f(x[n-1]);
integ+=(h/2)*(sum+y[0]+y[n-1]);
printf("FInal answer:%f",integ);
}
```

Output:



Experiment no. 8

Aim:

To solve the problem using implement simpson's rule.

Algorithm:

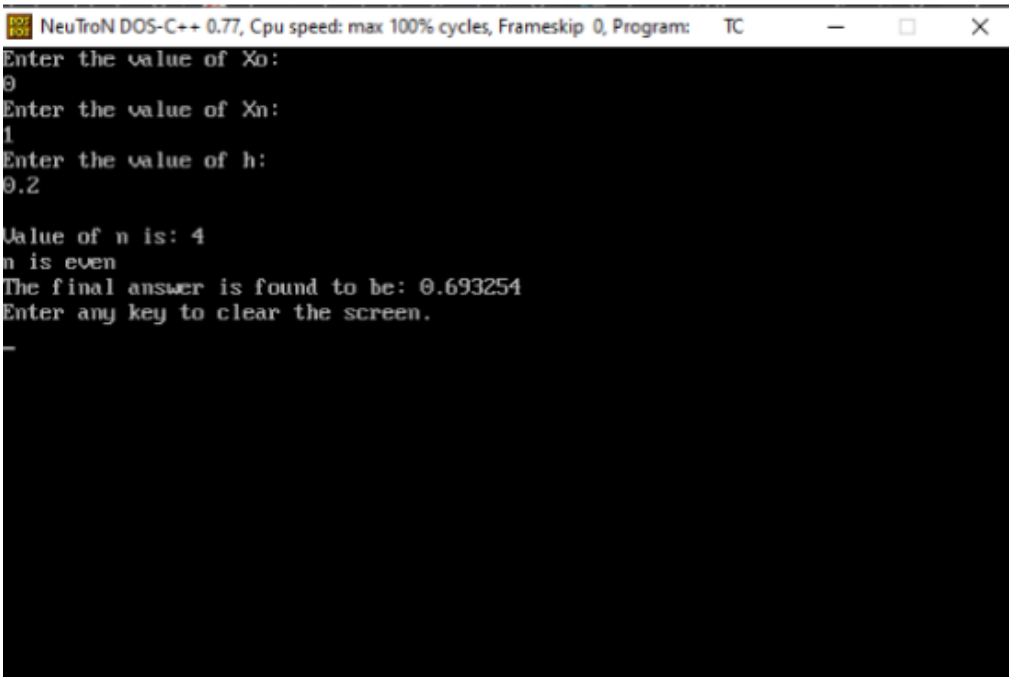
1. Start
2. Define a,b,n,h,z,sum = 0,Result.
3. Find h = (b-a)/n.
4. Define x[i],f[i].
5. Run loop to find x[i],f[i].
6. Run if conditional statements to modify the sum of odd and even indexed terms.
7. Compute Result from the formula
8. Result = h(f0+sum+f(n))/3.
9. Stop

Input:

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

float f(float x){
return (1/(1+x));
}

void main(){
int i,n;
float x0, xn, h, y[20], x[20], so, se, ans;
printf("Enter the value of Xo:\n");
scanf("%f", &x0);
printf("Enter the value of Xn:\n");
scanf("%f",&xn);
printf("Enter the value of h: \n");
scanf("%f",&h);
n=(xn-x0)/h;
printf("\nValue of n is: %d",n);
if (n%2==1){
printf("\nn is odd");
n=n+1;
}else{
printf("\nn is even");
}
h=(xn-x0)/n;
for (i=0;i<(n+1);i++){
x[i]=x[0]+(i*h);
y[i]=f(x[i]);
}
se=0;
so=0;
for (i=1;i<n;i++){
if (i%2==1){
so+=y[i];
```

```
}
else{
se+=y[i];
}
}
ans=(h/3)*(y[0]+y[n]+(4*so)+(2*se));
printf("\nThe final answer is found to be: %f",ans);
printf("\nEnter any key to clear the screen.\n");
getch();
clrscr();
}
```

Output:



Experiment no. 9

Aim:

To solve the problem using implement Euler's method.

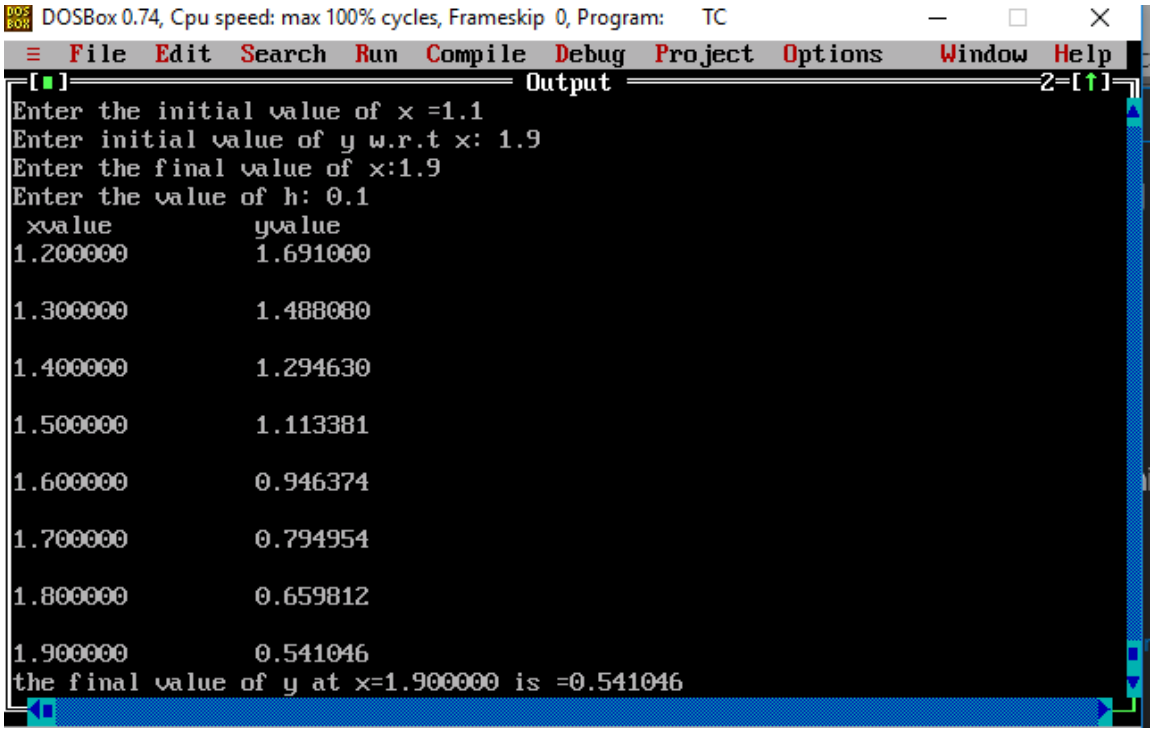
Algorithm:

1. Start
2. Declare f(x,y).
3. Read xo,yo,xn,h from the user(where xn is the final value of x).
4. Apply for(x=x0;x<xn;x+=h) for computing values of y by using formula- yn+1=yn+h*f(xn,yn).
5. Display the values of x and y on the screen.
6. Print the final value of y.
7. Stop

Input:

```
#include<stdio.h>
#include<math.h>
void main()
{
 double x0,y0,xn,h,x,y;
 double f(double x, double y);
 printf("Enter the initial value of x =");
 scanf("%lf",&x0);
 printf("Enter initial value of y w.r.t x: ");
 scanf("%lf",&y0);
printf("Enter the final value of x:");
 scanf("%lf",&xn);
 printf("Enter the value of h: ");
 scanf("%lf",&h);
 printf(" xvalue \t yvalue");
 for(x=x0;x<xn;x+=h)
 {
 y=y0+h*f(x,y0);
 printf("\n");
 printf("%lf \t %lf \n",x+h,y);
 y0=y;
 }
printf("the final value of y at x=%lf is =%lf",x,y);
}

double f(double x, double y)
{
 return (-(x*y));
}
```

Output:


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
File Edit Search Run Compile Debug Project Options Window Help
Output
Enter the initial value of x =1.1
Enter initial value of y w.r.t x: 1.9
Enter the final value of x:1.9
Enter the value of h: 0.1
xvalue yvalue
1.200000 1.691000
1.300000 1.488080
1.400000 1.294630
1.500000 1.113381
1.600000 0.946374
1.700000 0.794954
1.800000 0.659812
1.900000 0.541046
the final value of y at x=1.900000 is =0.541046

Experiment no. 10

Aim:

To solve the problem using implement Gauss seidel Iteration method.

Algorithm:

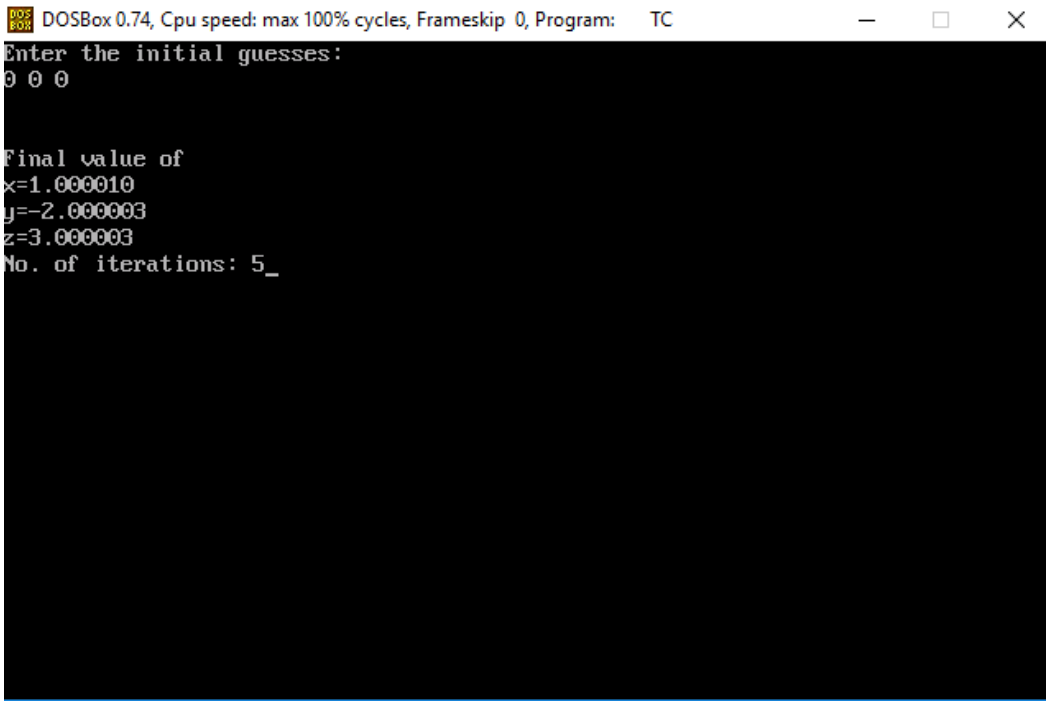
1. Start
2. Define function f(x,y,z),s(x,y,z),t(x,y,z)
3. Choose initial guesses x0,y0,z0 .
4. Choose pre-specified tolerable error.
5. Run do while loop changing the values of x1, y1, z1 by using the assumed given values/guesses.
6. While |dx-x1| > error,|dy-y1|>error,|dz-z1|>error then goto (5) otherwise goto (7).
7. Display x1,y1,z1 as root.
8. Stop

Input:

```
/* Program to implement Gauss Scidel Method */
#include<stdio.h>
#include<conio.h>
#include<math.h>

float f(float x, float y, float z)
{
x=((9-(2*y)-z)/10);
return x;
}

float s(float x,float y,float z)
{
y=((-44-(2*x)+(2*z))/20);
return y;
}

float t(float x,float y,float z)
{
z=((-22-(2*x)+(3*y))/(-10));
return z;
}

void main()
{
clrscr();
float x0,y0,z0,x1=0,y1=0,z1=0,tempx,tempy,tempz,acc=0.0001;
int i=0;
printf("Enter the initial guesses:\n");
scanf("%f %f %f",&x0, &y0, &z0);
do{
tempx=x1;
tempy=y1;
tempz=z1;
x1=f(x0,y0,z0);
y1=s(x1,y0,z0);
z1=t(x1,y1,z0);
```

```
i++;
x0=x1;
y0=y1;
z0=z1;
}
while(fabs(tempx-x1)>acc&&fabs(tempy-y1)>acc&&fabs(tempz-z1)>acc);

printf("\n\nFinal value of\n");
printf("x=%f \ny=%f \nz=%f \n",x1,y1,z1);
printf("No. of iterations: %d",i);
getch();
}
```

Output:



Experiment no. 11

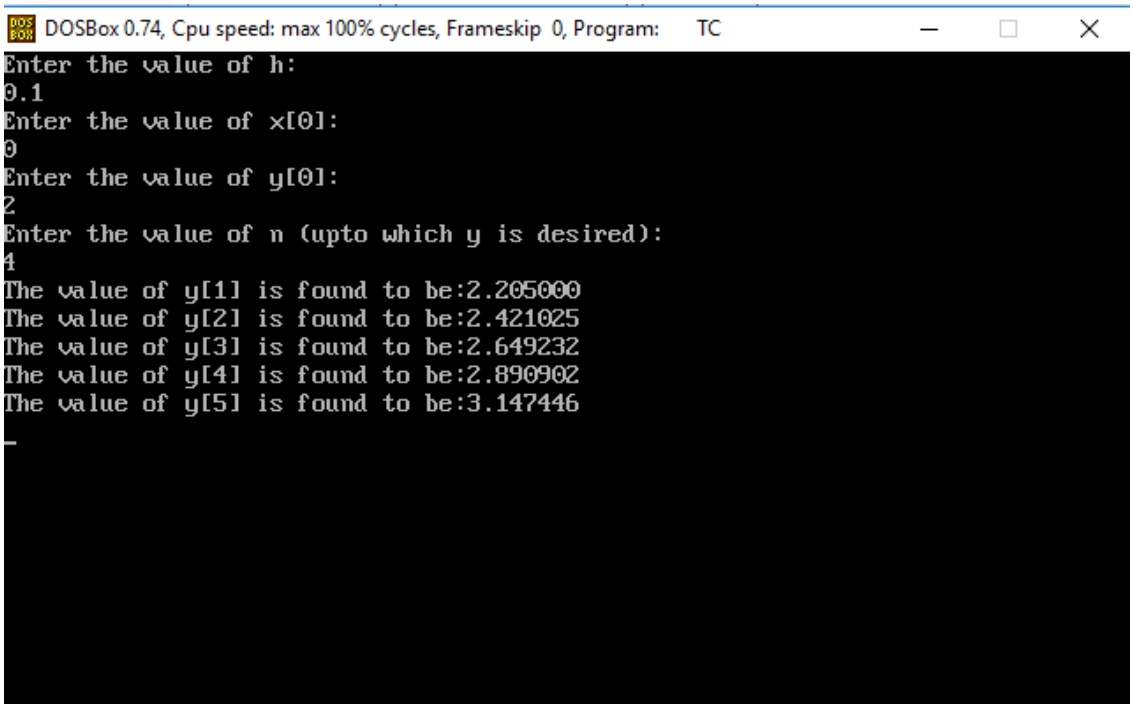
Aim:

To solve the problem using implement Runge-Kutta method.

Input:

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

float f(float x,float y){
float a=y-x;
return a;
}

void main(){
clrscr();
float h,k1,k2,k3,k4,a,b;
int n,i;
float x[20], y[20];
printf("Enter the value of h:\n");
scanf("%f",&h);
printf("Enter the value of x[0]:\n");
scanf("%f",&a);
x[0]=a;
printf("Enter the value of y[0]:\n");
scanf("%f",&b);
y[0]=b;
printf("Enter the value of n (upto which y is desired):\n");
scanf("%d",&n);
for (i=0;i<(n+1);i++){
k1=h*f(x[i],y[i]);
x[i+1]=x[i]+h;
k2=h*f(x[i+1],(y[i]+k1));
y[i+1]=y[i]+(0.5*(k1+k2));
printf("The value of y[%d] is found to be:%f\n",(i+1),y[i+1]);
}
getch();

}
```

Output:


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
Enter the value of h:
0.1
Enter the value of x[0]:
0
Enter the value of y[0]:
2
Enter the value of n (upto which y is desired):
4
The value of y[1] is found to be:2.205000
The value of y[2] is found to be:2.421025
The value of y[3] is found to be:2.649232
The value of y[4] is found to be:2.890902
The value of y[5] is found to be:3.147446