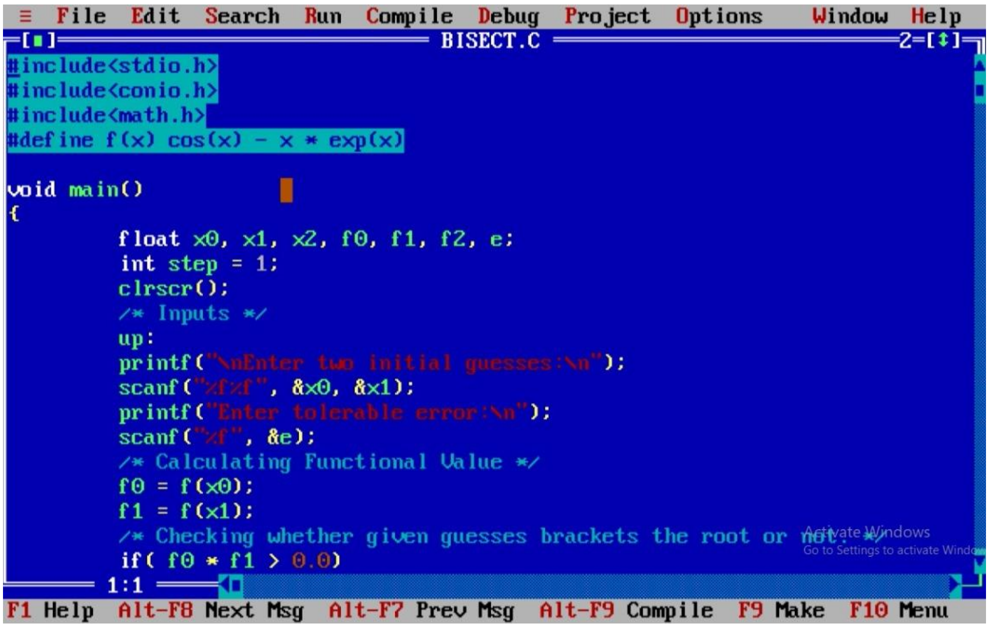
Name: Harsh Nair

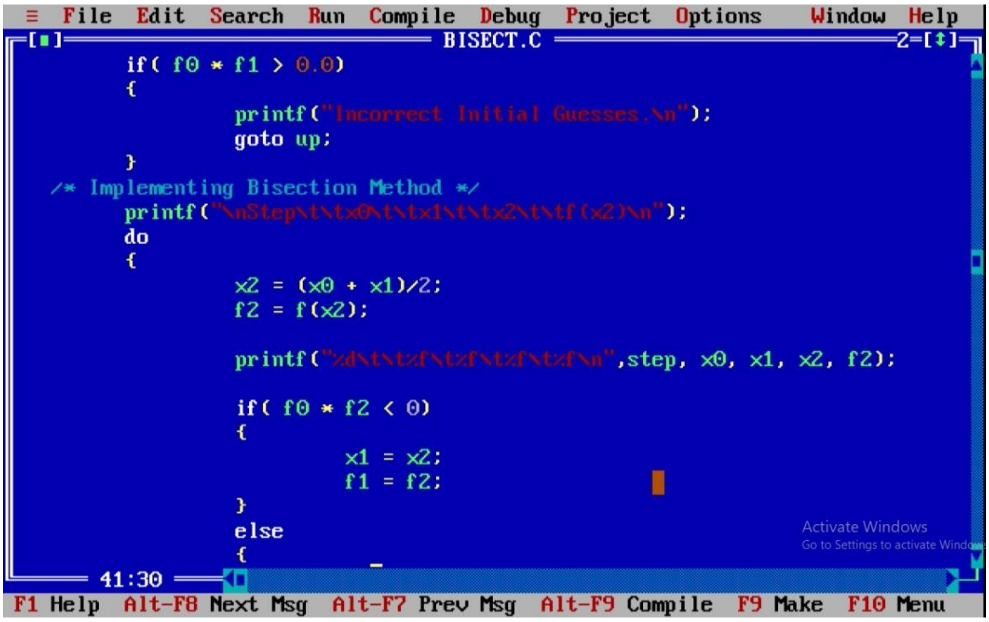
Roll No. FCS2122074

Calculus Practicals

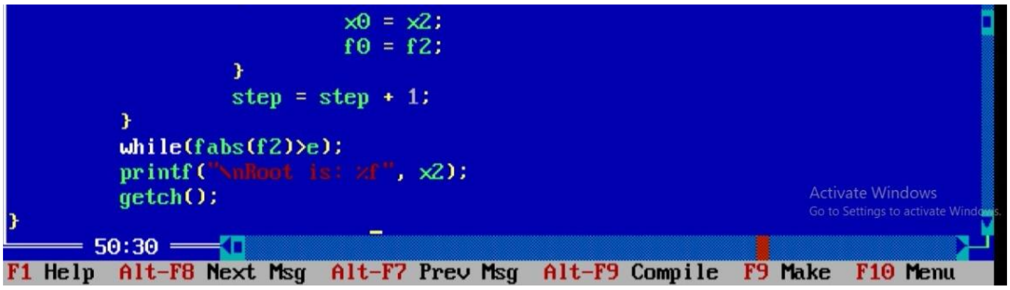
Bisection Method:


≡ File Edit Search Run Compile Debug Project Options Window Help
[■] BISECT.C 2=[↕]
1:1
Activate Windows
Go to Settings to activate Windows
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
#define f(x) cos(x) - x * exp(x)

void main()
{
        float x0, x1, x2, f0, f1, f2, e;
        int step = 1;
        clrscr();
        /* Inputs */
        up:
        printf("\nEnter two initial guesses:\n");
        scanf("%f%f", &x0, &x1);
        printf("Enter tolerable error:\n");
        scanf("%f", &e);
        /* Calculating Functional Value */
        f0 = f(x0);
        f1 = f(x1);
        /* Checking whether given guesses brackets the root or not. */
        if( f0 * f1 > 0.0)
```

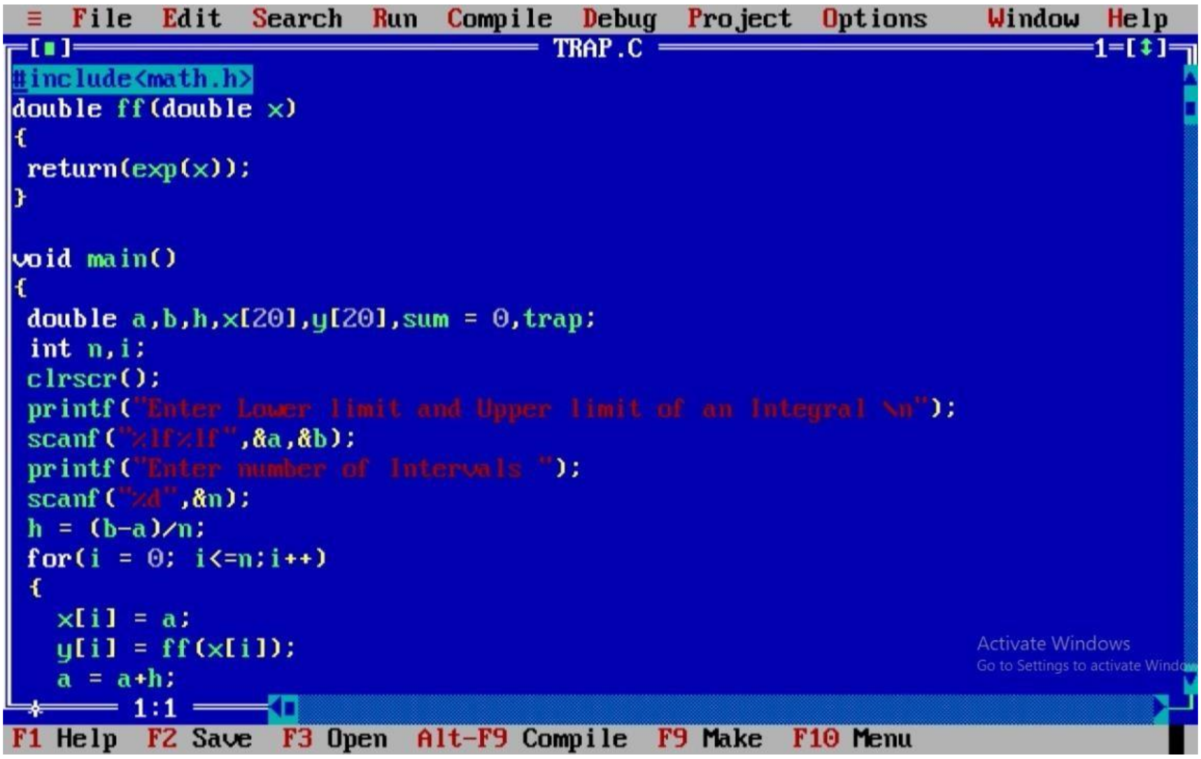

≡ File Edit Search Run Compile Debug Project Options Window Help
[■] BISECT.C 2=[↕]
41:30
Activate Windows
Go to Settings to activate Windows
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

```
        if( f0 * f1 > 0.0)
        {
                printf("Incorrect Initial Guesses.\n");
                goto up;
        }
    /* Implementing Bisection Method */
        printf("\nStep\t\tx0\t\tx1\t\tx2\t\tf(x2)\n");
        do
        {
                x2 = (x0 + x1)/2;
                f2 = f(x2);

                printf("%d\t\t%f\t%f\t%f\t%f\n",step, x0, x1, x2, f2);

                if( f0 * f2 < 0)
                {
                        x1 = x2;
                        f1 = f2;
                }
                else
                {
```

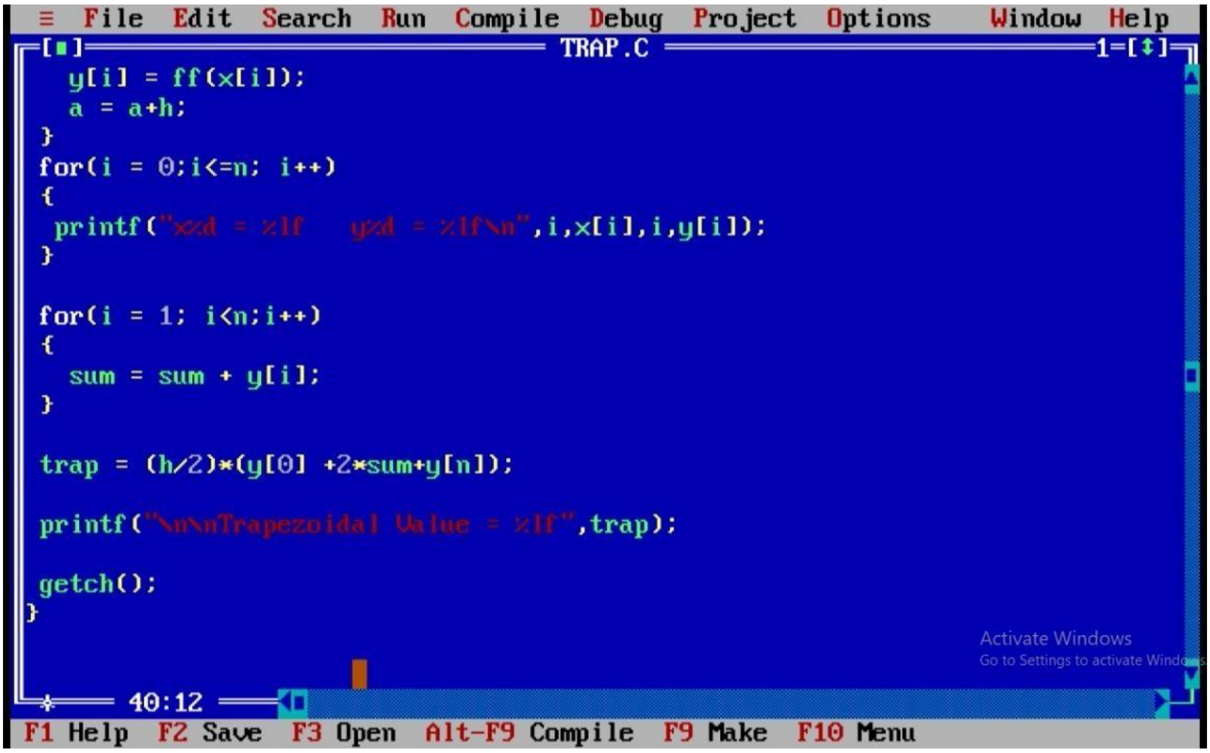

50:30
Activate Windows
Go to Settings to activate Windows.
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

```
                        x0 = x2;
                        f0 = f2;
                }
                step = step + 1;
        }
        while(fabs(f2)>e);
        printf("\nRoot is: %f", x2);
        getch();
}
```

Trapizoidal:

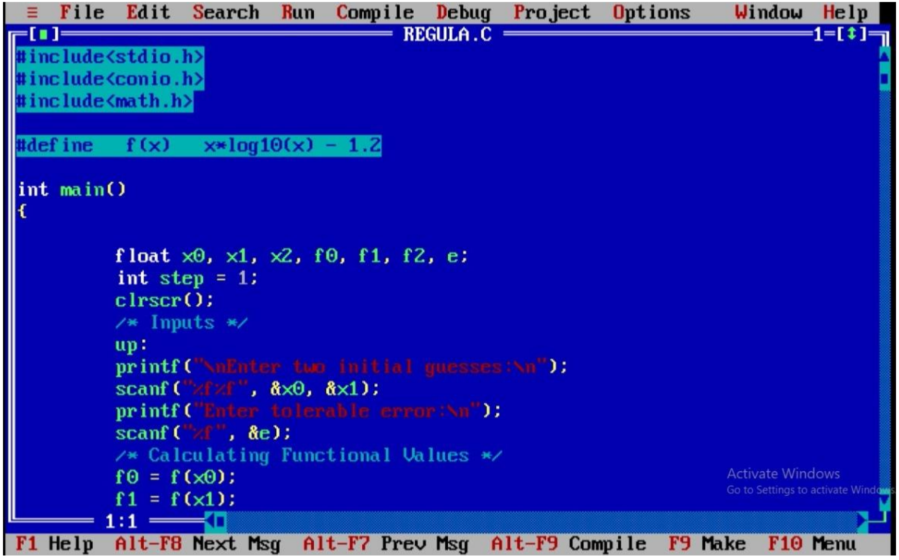

≡ File Edit Search Run Compile Debug Project Options Window Help
[■] TRAP.C 1=[↕]
1:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu
Activate Windows
Go to Settings to activate Windows

```
#include<math.h>
double ff(double x)
{
 return(exp(x));
}

void main()
{
 double a,b,h,x[20],y[20],sum = 0,trap;
 int n,i;
 clrscr();
 printf("Enter Lower limit and Upper limit of an Integral \n");
 scanf("%lf%lf",&a,&b);
 printf("Enter number of Intervals ");
 scanf("%d",&n);
 h = (b-a)/n;
 for(i = 0; i<=n;i++)
 {
   x[i] = a;
   y[i] = ff(x[i]);
   a = a+h;
```

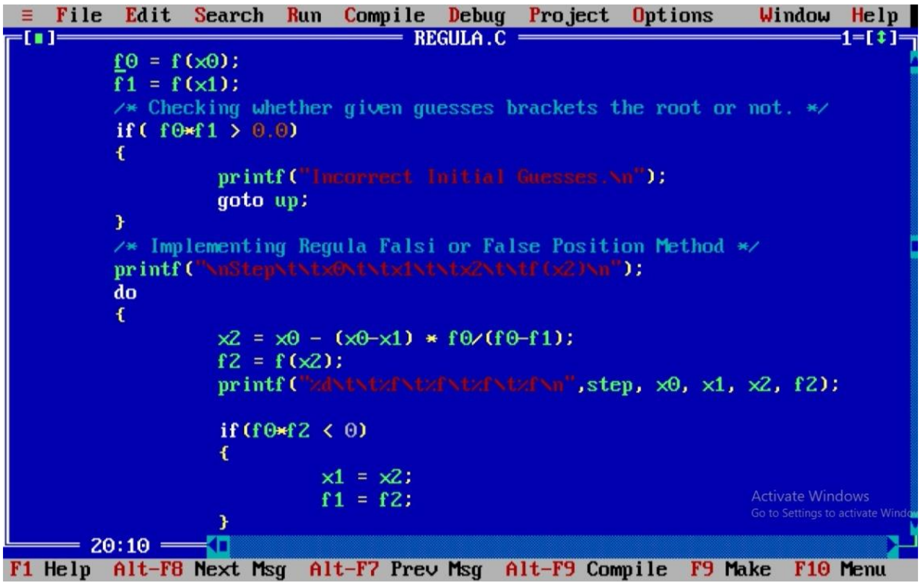

≡ File Edit Search Run Compile Debug Project Options Window Help
[■] TRAP.C 1=[↕]
40:12
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu
Activate Windows
Go to Settings to activate Windows

```
   y[i] = ff(x[i]);
   a = a+h;
 }
 for(i = 0;i<=n; i++)
 {
  printf("x%d = %lf    y%d = %lf\n",i,x[i],i,y[i]);
 }

 for(i = 1; i<n;i++)
 {
   sum = sum + y[i];
 }

 trap = (h/2)*(y[0] +2*sum+y[n]);

 printf("\n\nTrapezoidal Value = %lf",trap);

 getch();
}
```

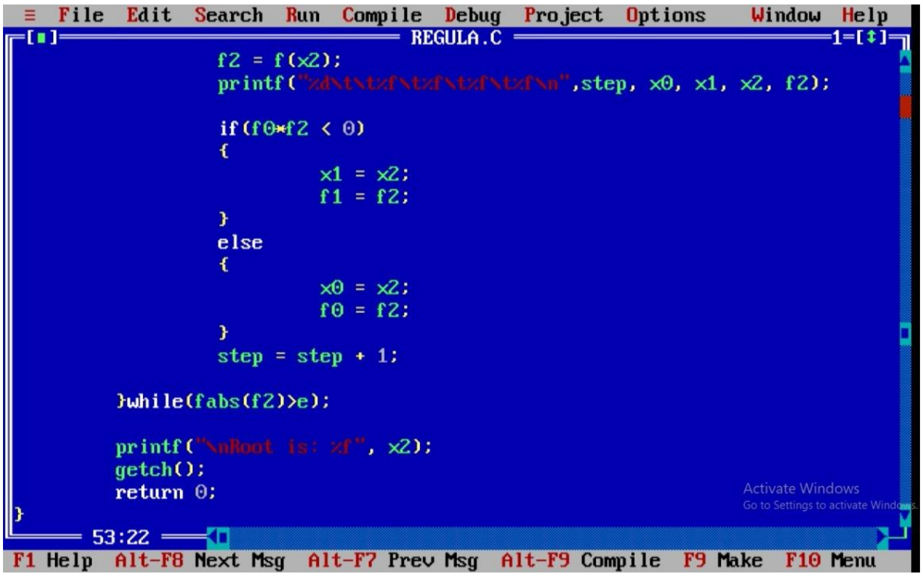
Regula Falsi Method:


≡ File Edit Search Run Compile Debug Project Options Window Help
REGULA.C
1=[↕]
1:1
Activate Windows
Go to Settings to activate Windows.
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

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

#define   f(x)   x*log10(x) - 1.2

int main()
{

        float x0, x1, x2, f0, f1, f2, e;
        int step = 1;
        clrscr();
        /* Inputs */
        up:
        printf("\nEnter two initial guesses:\n");
        scanf("%f%f", &x0, &x1);
        printf("Enter tolerable error:\n");
        scanf("%f", &e);
        /* Calculating Functional Values */
        f0 = f(x0);
        f1 = f(x1);
```

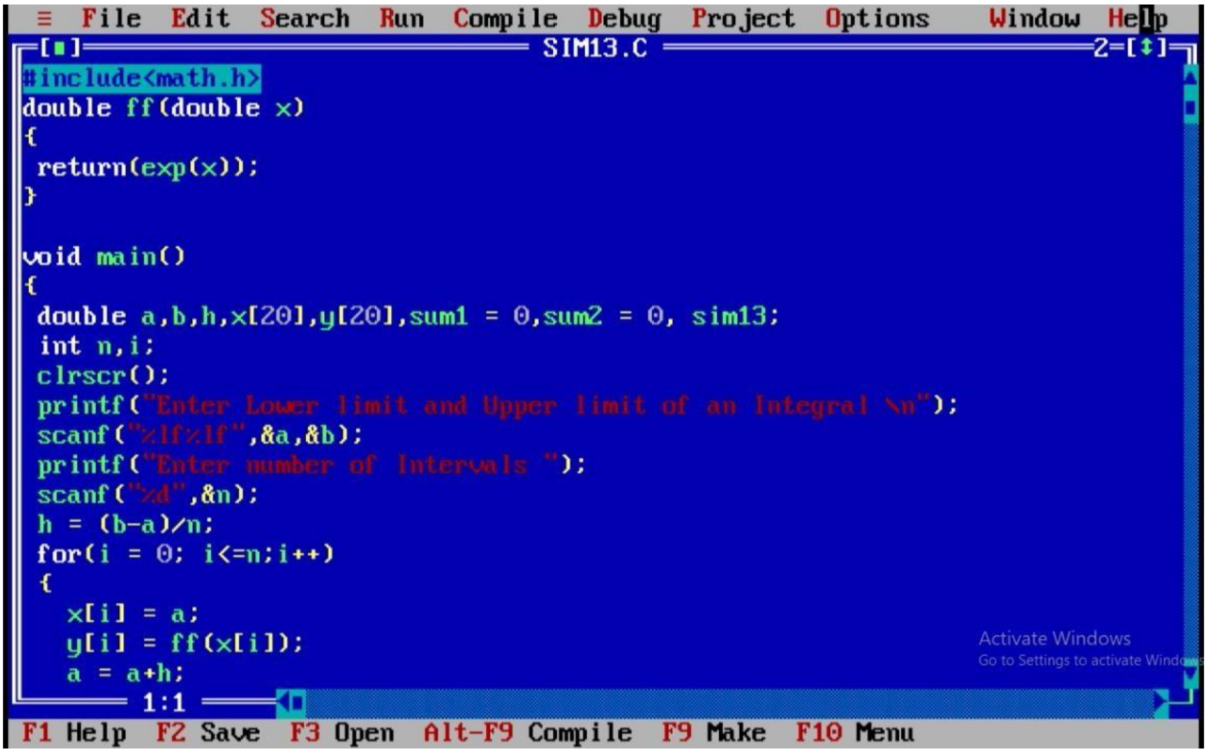

≡ File Edit Search Run Compile Debug Project Options Window Help
REGULA.C
1=[↕]
20:10
Activate Windows
Go to Settings to activate Windows.
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

```c
        f0 = f(x0);
        f1 = f(x1);
        /* Checking whether given guesses brackets the root or not. */
        if( f0*f1 > 0.0)
        {
                printf("Incorrect Initial Guesses.\n");
                goto up;
        }
        /* Implementing Regula Falsi or False Position Method */
        printf("\nStep\t\tx0\t\tx1\t\tx2\t\tf(x2)\n");
        do
        {
                x2 = x0 - (x0-x1) * f0/(f0-f1);
                f2 = f(x2);
                printf("%d\t\t%f\t%f\t%f\t%f\n",step, x0, x1, x2, f2);

                if(f0*f2 < 0)
                {
                        x1 = x2;
                        f1 = f2;
                }
```

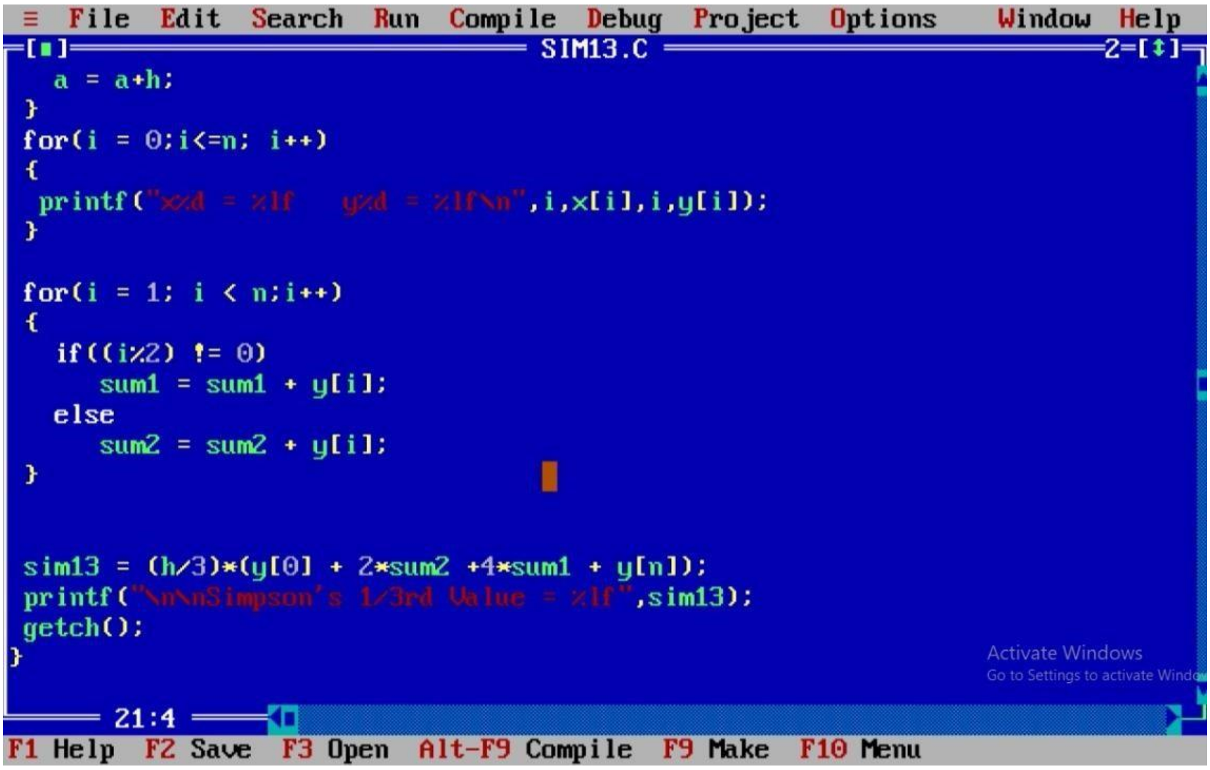

≡ File Edit Search Run Compile Debug Project Options Window Help
REGULA.C
1=[↕]
53:22
Activate Windows
Go to Settings to activate Windows.
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

```c
                f2 = f(x2);
                printf("%d\t\t%f\t%f\t%f\t%f\n",step, x0, x1, x2, f2);

                if(f0*f2 < 0)
                {
                        x1 = x2;
                        f1 = f2;
                }
                else
                {
                        x0 = x2;
                        f0 = f2;
                }
                step = step + 1;

        }while(fabs(f2)>e);

        printf("\nRoot is: %f", x2);
        getch();
        return 0;
}
```

Simpsons 1/3 rule:

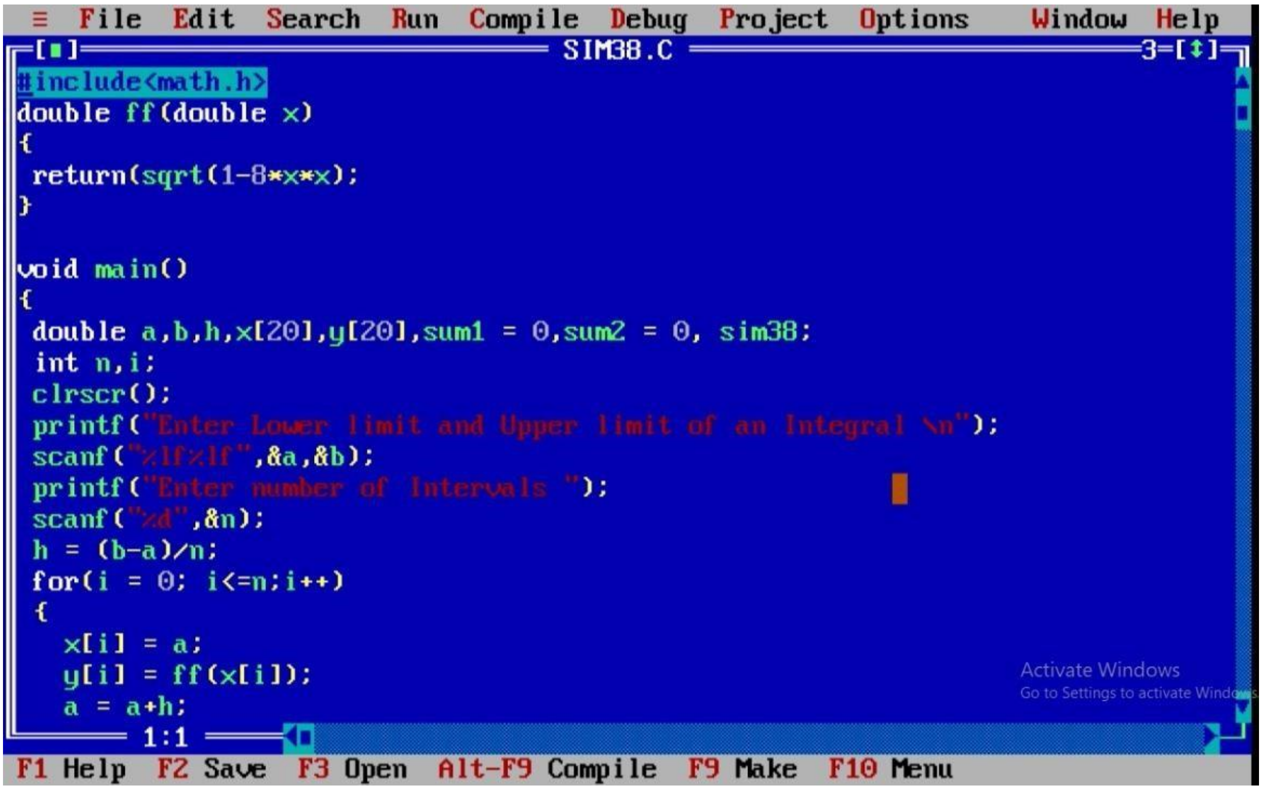

≡ File Edit Search Run Compile Debug Project Options Window Help
[■] SIM13.C 2=[↕]
1:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu
Activate Windows
Go to Settings to activate Windows

```
#include<math.h>
double ff(double x)
{
 return(exp(x));
}

void main()
{
 double a,b,h,x[20],y[20],sum1 = 0,sum2 = 0, sim13;
 int n,i;
 clrscr();
 printf("Enter Lower limit and Upper limit of an Integral \n");
 scanf("%lf%lf",&a,&b);
 printf("Enter number of Intervals ");
 scanf("%d",&n);
 h = (b-a)/n;
 for(i = 0; i<=n;i++)
 {
   x[i] = a;
   y[i] = ff(x[i]);
   a = a+h;
```

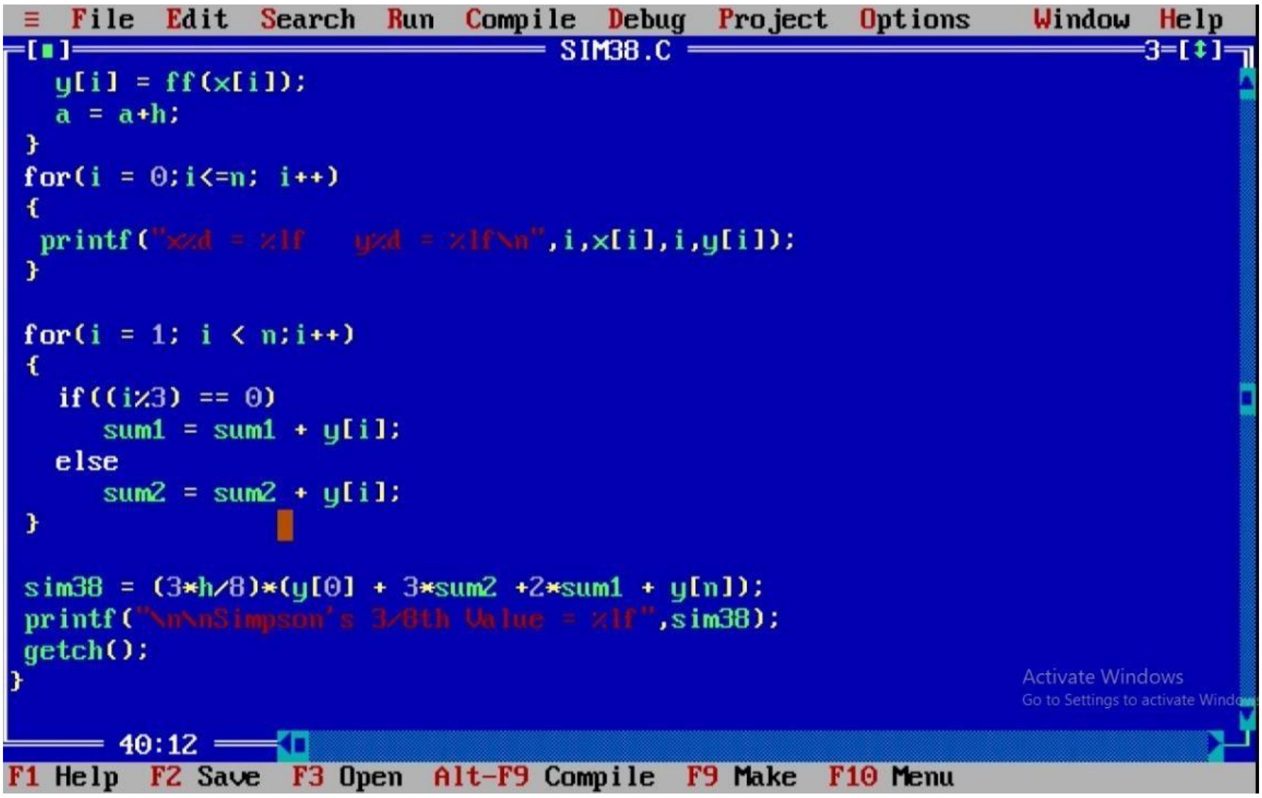

≡ File Edit Search Run Compile Debug Project Options Window Help
[■] SIM13.C 2=[↕]
21:4
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu
Activate Windows
Go to Settings to activate Windows

```
   a = a+h;
 }
 for(i = 0;i<=n; i++)
 {
  printf("x%d = %lf    y%d = %lf\n",i,x[i],i,y[i]);
 }

 for(i = 1; i < n;i++)
 {
   if((i%2) != 0)
      sum1 = sum1 + y[i];
   else
      sum2 = sum2 + y[i];
 }


 sim13 = (h/3)*(y[0] + 2*sum2 +4*sum1 + y[n]);
 printf("\n\nSimpson's 1/3rd Value = %lf",sim13);
 getch();
}
```

Simpsons 3/8 rule:

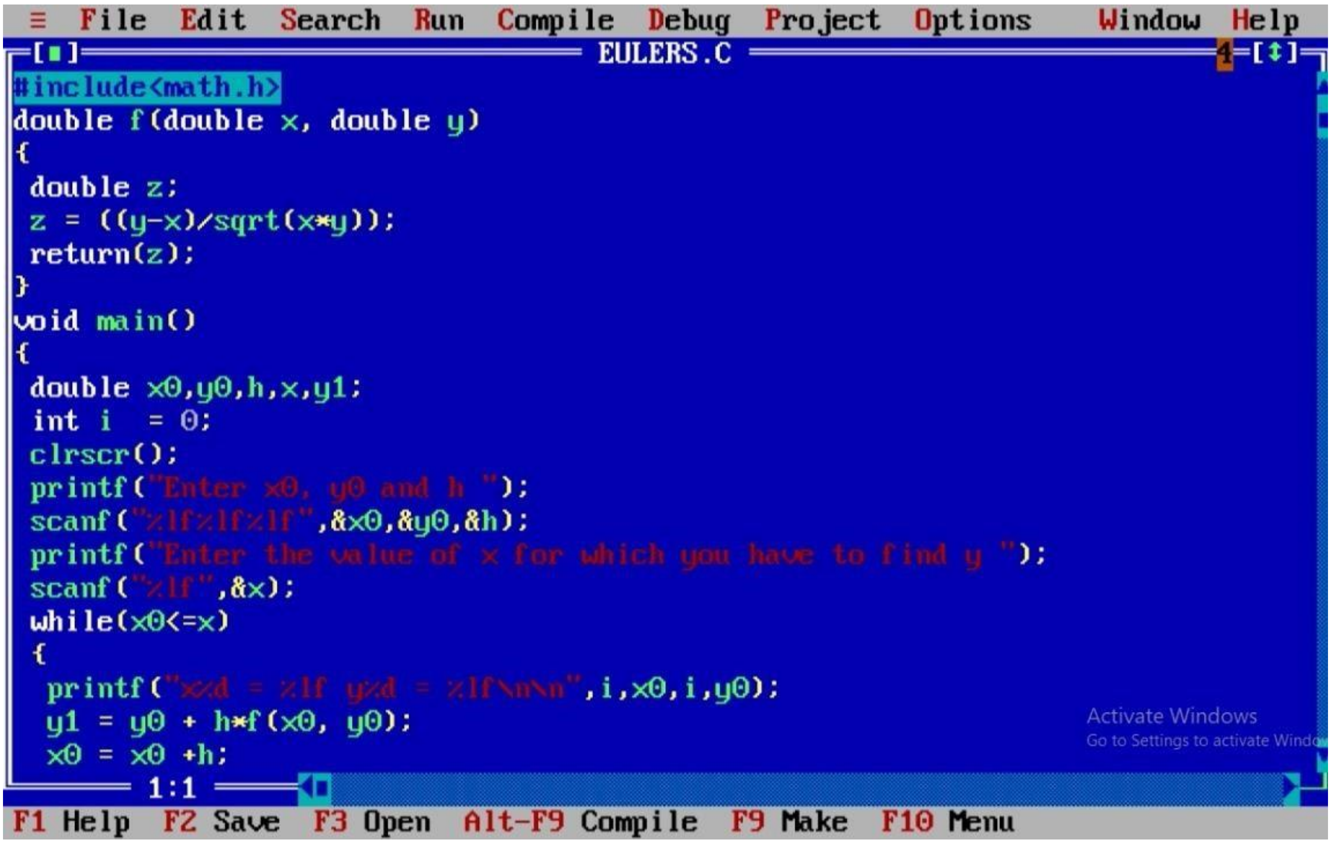

≡ File Edit Search Run Compile Debug Project Options Window Help
[■] SIM38.C 3=[↕]
1:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu
Activate Windows
Go to Settings to activate Windows.

```
#include<math.h>
double ff(double x)
{
 return(sqrt(1-8*x*x);
}

void main()
{
 double a,b,h,x[20],y[20],sum1 = 0,sum2 = 0, sim38;
 int n,i;
 clrscr();
 printf("Enter Lower limit and Upper limit of an Integral \n");
 scanf("%lf%lf",&a,&b);
 printf("Enter number of Intervals ");
 scanf("%d",&n);
 h = (b-a)/n;
 for(i = 0; i<=n;i++)
  {
    x[i] = a;
    y[i] = ff(x[i]);
    a = a+h;
```

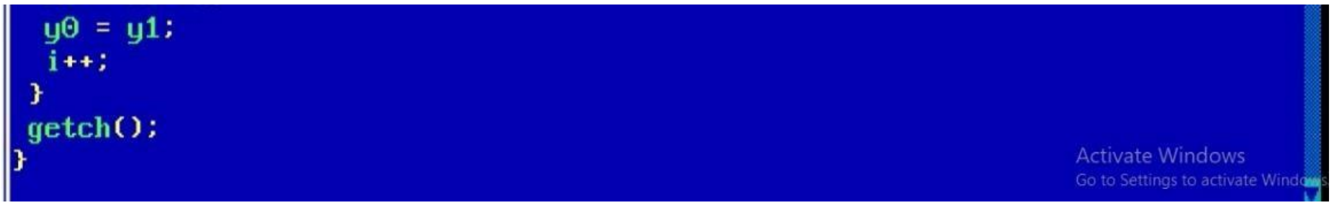

≡ File Edit Search Run Compile Debug Project Options Window Help
[■] SIM38.C 3=[↕]
40:12
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu
Activate Windows
Go to Settings to activate Windows.

```
   y[i] = ff(x[i]);
   a = a+h;
 }
 for(i = 0;i<=n; i++)
 {
  printf("x%d = %lf    y%d = %lf\n",i,x[i],i,y[i]);
 }

 for(i = 1; i < n;i++)
 {
   if((i%3) == 0)
      sum1 = sum1 + y[i];
   else
      sum2 = sum2 + y[i];
 }

 sim38 = (3*h/8)*(y[0] + 3*sum2 +2*sum1 + y[n]);
 printf("\n\nSimpson's 3/8th Value = %lf",sim38);
 getch();
}
```

Euler’s rule:

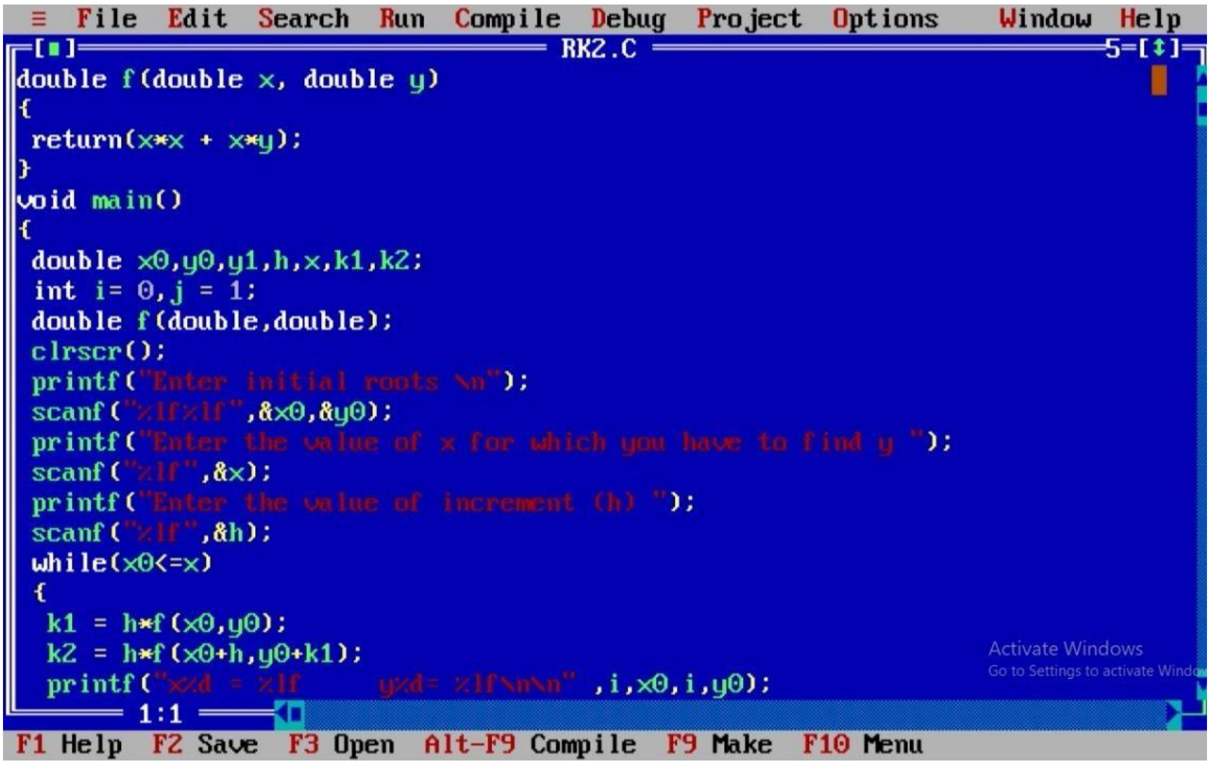

≡ File Edit Search Run Compile Debug Project Options Window Help
[■] EULERS.C 4 [↕]
1:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu
Activate Windows
Go to Settings to activate Windows

```
#include<math.h>
double f(double x, double y)
{
 double z;
 z = ((y-x)/sqrt(x*y));
 return(z);
}
void main()
{
 double x0,y0,h,x,y1;
 int i  = 0;
 clrscr();
 printf("Enter x0, y0 and h ");
 scanf("%lf%lf%lf",&x0,&y0,&h);
 printf("Enter the value of x for which you have to find y ");
 scanf("%lf",&x);
 while(x0<=x)
 {
  printf("x%d = %lf y%d = %lf\n\n",i,x0,i,y0);
  y1 = y0 + h*f(x0, y0);
  x0 = x0 +h;
```

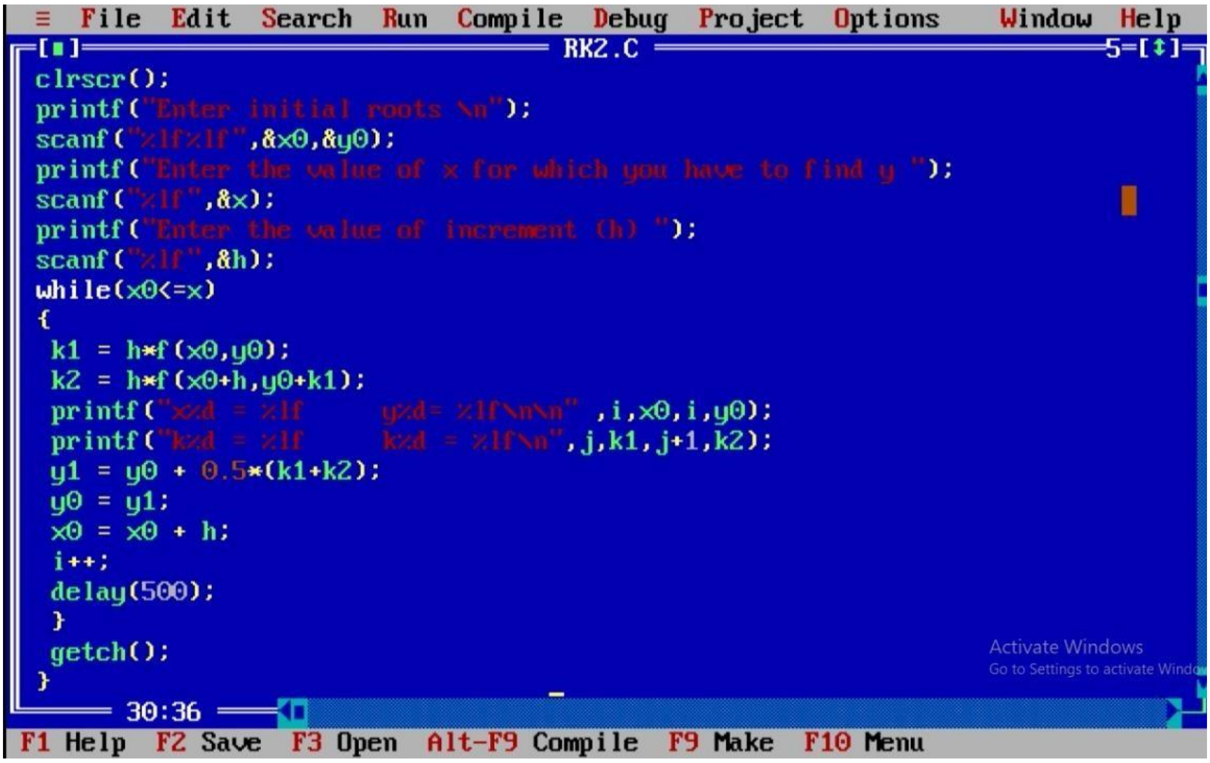


```
  y0 = y1;
  i++;
 }
 getch();
}
```

RK-2 (Range Kutta):

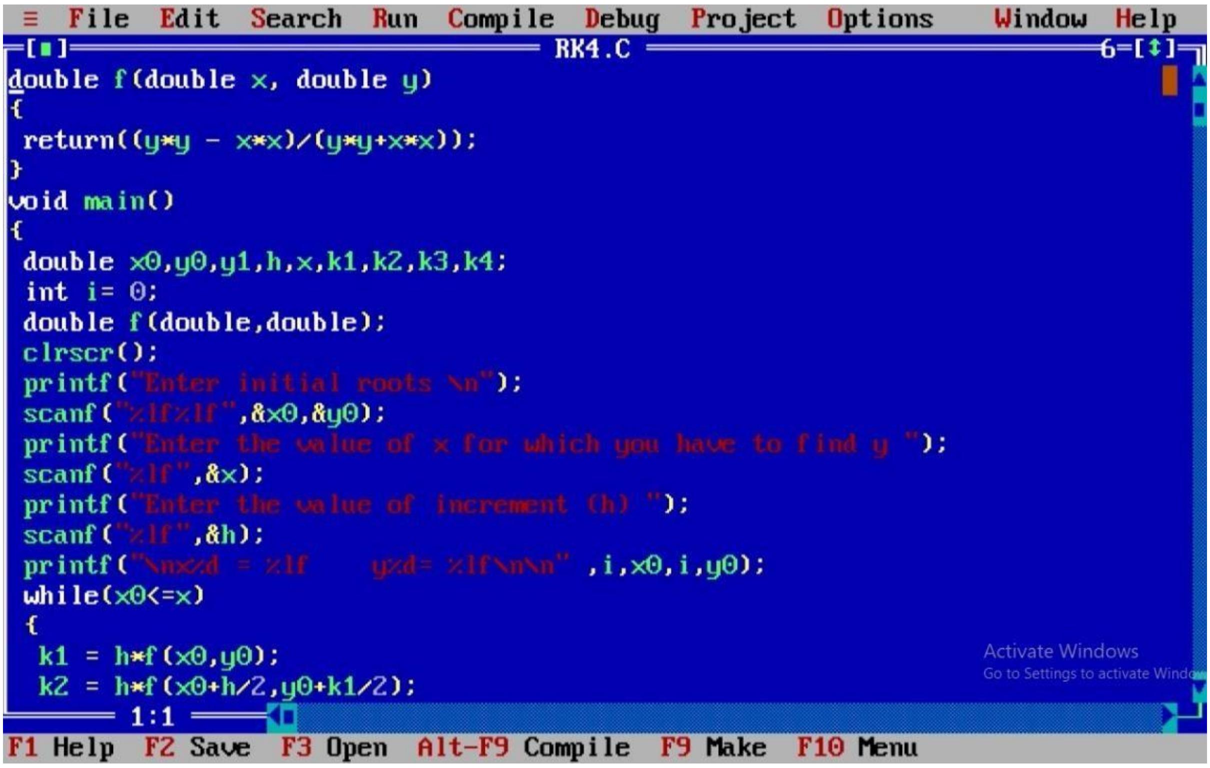

≡ File Edit Search Run Compile Debug Project Options Window Help
[■] RK2.C 5=[↕]
Activate Windows
Go to Settings to activate Windo
1:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

```
double f(double x, double y)
{
 return(x*x + x*y);
}
void main()
{
 double x0,y0,y1,h,x,k1,k2;
 int i= 0,j = 1;
 double f(double,double);
 clrscr();
 printf("Enter initial roots \n");
 scanf("%lf%lf",&x0,&y0);
 printf("Enter the value of x for which you have to find y ");
 scanf("%lf",&x);
 printf("Enter the value of increment (h) ");
 scanf("%lf",&h);
 while(x0<=x)
 {
  k1 = h*f(x0,y0);
  k2 = h*f(x0+h,y0+k1);
  printf("x%d = %lf        y%d= %lf\n\n" ,i,x0,i,y0);
```

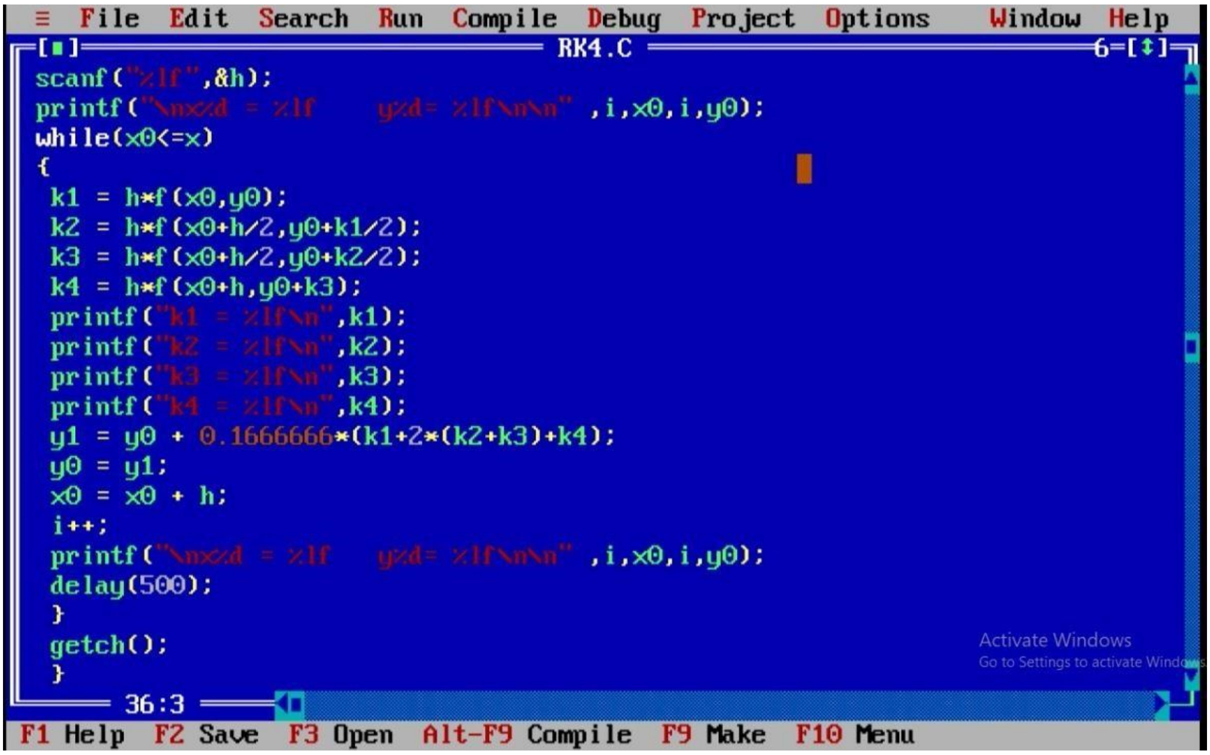

≡ File Edit Search Run Compile Debug Project Options Window Help
[■] RK2.C 5=[↕]
Activate Windows
Go to Settings to activate Windo
30:36
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

```
 clrscr();
 printf("Enter initial roots \n");
 scanf("%lf%lf",&x0,&y0);
 printf("Enter the value of x for which you have to find y ");
 scanf("%lf",&x);
 printf("Enter the value of increment (h) ");
 scanf("%lf",&h);
 while(x0<=x)
 {
  k1 = h*f(x0,y0);
  k2 = h*f(x0+h,y0+k1);
  printf("x%d = %lf        y%d= %lf\n\n" ,i,x0,i,y0);
  printf("k%d = %lf        k%d = %lf\n",j,k1,j+1,k2);
  y1 = y0 + 0.5*(k1+k2);
  y0 = y1;
  x0 = x0 + h;
  i++;
  delay(500);
  }
  getch();
 }
```

RK-4:

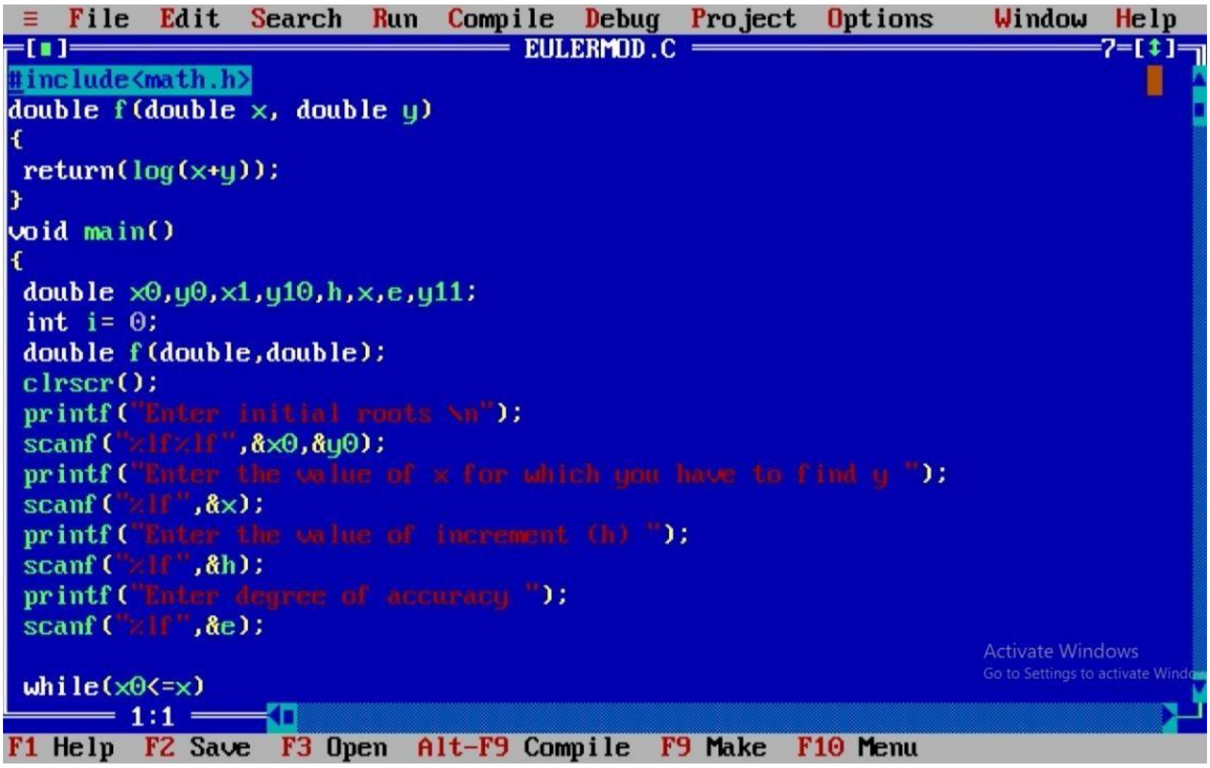

≡ File Edit Search Run Compile Debug Project Options Window Help
[■] RK4.C 6=[↕]
1:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

```
double f(double x, double y)
{
 return((y*y - x*x)/(y*y+x*x));
}
void main()
{
 double x0,y0,y1,h,x,k1,k2,k3,k4;
 int i= 0;
 double f(double,double);
 clrscr();
 printf("Enter initial roots \n");
 scanf("%lf%lf",&x0,&y0);
 printf("Enter the value of x for which you have to find y ");
 scanf("%lf",&x);
 printf("Enter the value of increment (h) ");
 scanf("%lf",&h);
 printf("\nx%d = %lf     y%d= %lf\n\n" ,i,x0,i,y0);
 while(x0<=x)
 {
  k1 = h*f(x0,y0);
  k2 = h*f(x0+h/2,y0+k1/2);
```

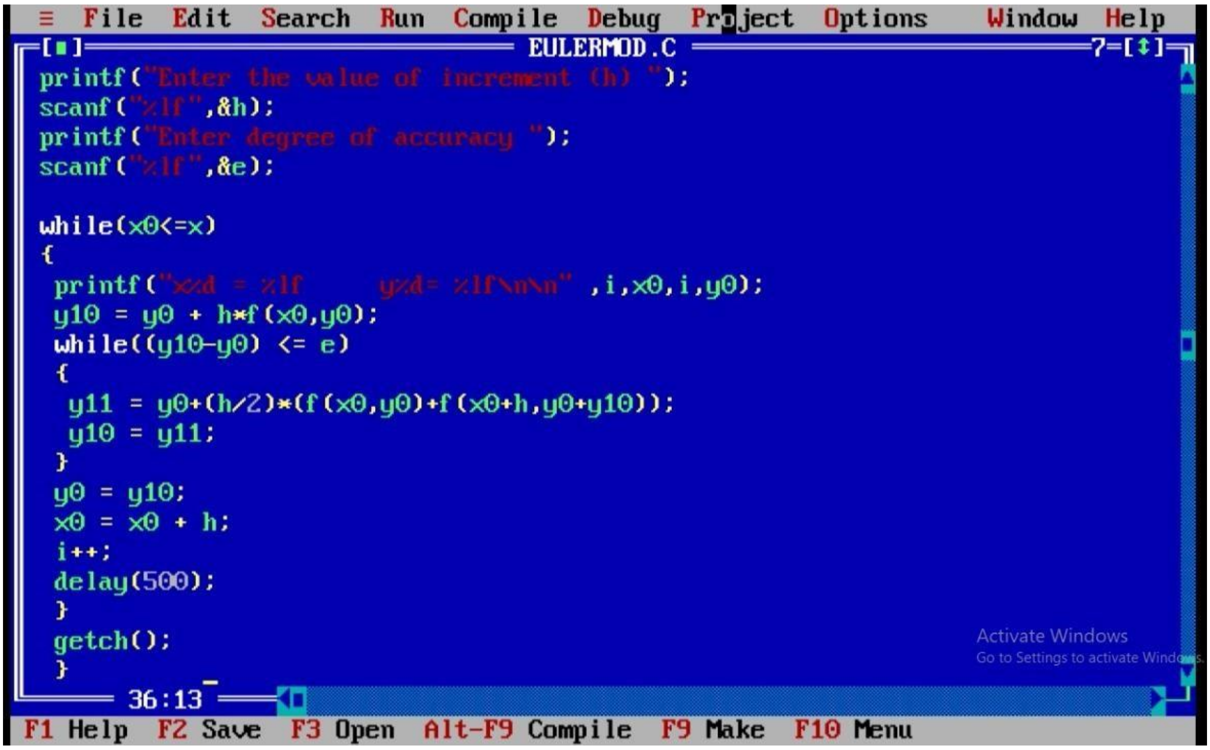

≡ File Edit Search Run Compile Debug Project Options Window Help
[■] RK4.C 6=[↕]
36:3
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

```
 scanf("%lf",&h);
 printf("\nx%d = %lf     y%d= %lf\n\n" ,i,x0,i,y0);
 while(x0<=x)
 {
  k1 = h*f(x0,y0);
  k2 = h*f(x0+h/2,y0+k1/2);
  k3 = h*f(x0+h/2,y0+k2/2);
  k4 = h*f(x0+h,y0+k3);
  printf("k1 = %lf\n",k1);
  printf("k2 = %lf\n",k2);
  printf("k3 = %lf\n",k3);
  printf("k4 = %lf\n",k4);
  y1 = y0 + 0.1666666*(k1+2*(k2+k3)+k4);
  y0 = y1;
  x0 = x0 + h;
  i++;
  printf("\nx%d = %lf     y%d= %lf\n\n" ,i,x0,i,y0);
  delay(500);
  }
  getch();
  }
```

Euler’s Modified:

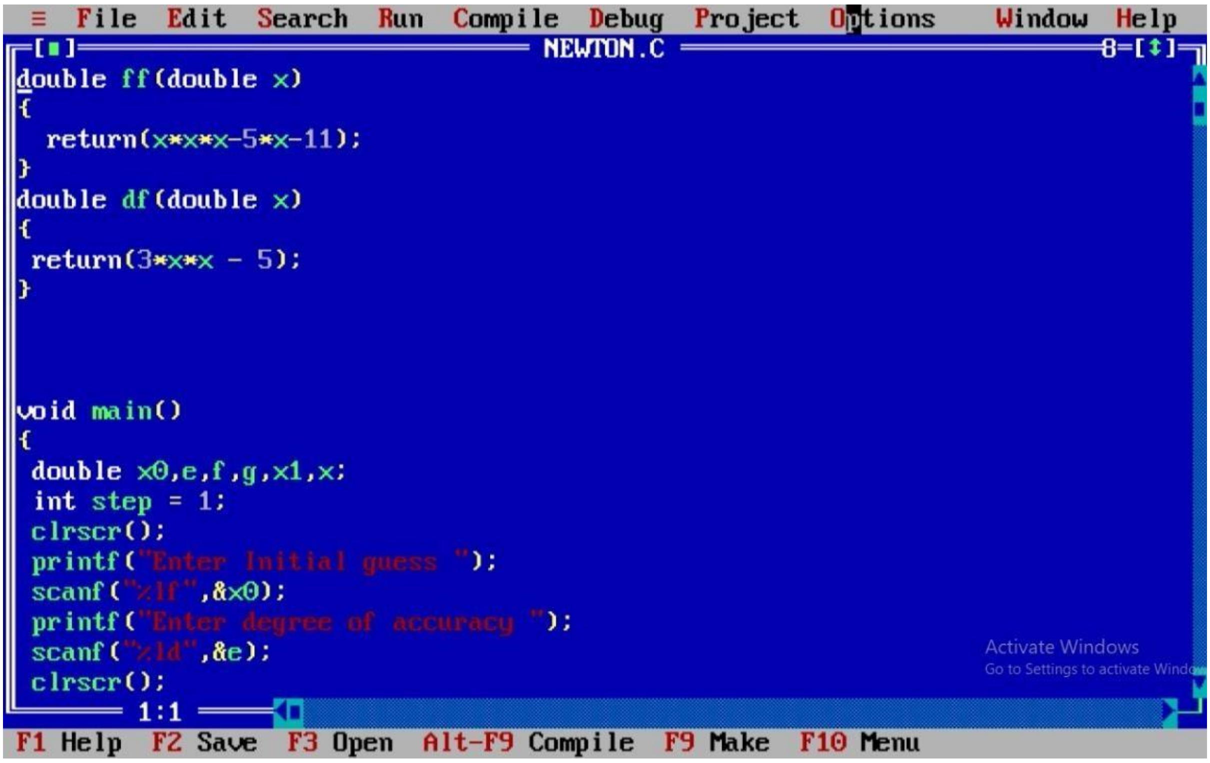

≡ File Edit Search Run Compile Debug Project Options Window Help
EULERMOD.C
7
1:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu
Activate Windows
Go to Settings to activate Windo

```
#include<math.h>
double f(double x, double y)
{
 return(log(x+y));
}
void main()
{
 double x0,y0,x1,y10,h,x,e,y11;
 int i= 0;
 double f(double,double);
 clrscr();
 printf("Enter initial roots \n");
 scanf("%lf%lf",&x0,&y0);
 printf("Enter the value of x for which you have to find y ");
 scanf("%lf",&x);
 printf("Enter the value of increment (h) ");
 scanf("%lf",&h);
 printf("Enter degree of accuracy ");
 scanf("%lf",&e);

 while(x0<=x)
```

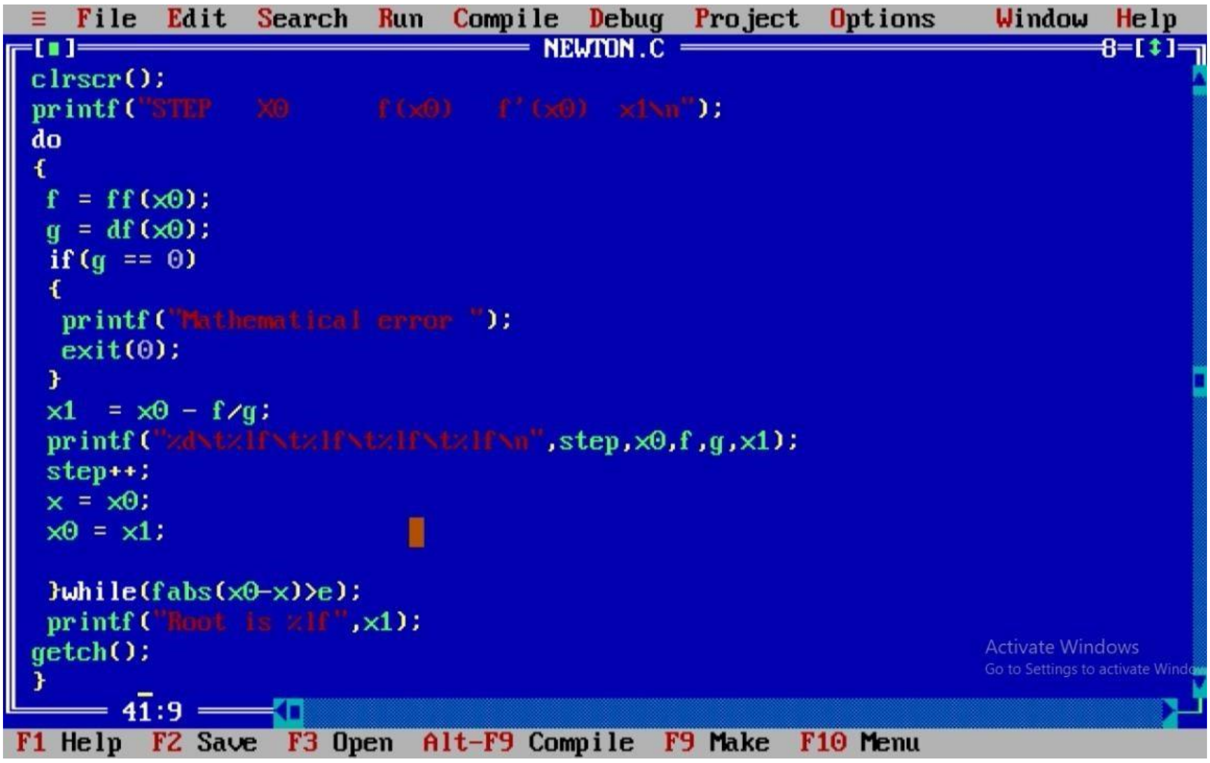

≡ File Edit Search Run Compile Debug Project Options Window Help
EULERMOD.C
7
36:13
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu
Activate Windows
Go to Settings to activate Windows.

```
 printf("Enter the value of increment (h) ");
 scanf("%lf",&h);
 printf("Enter degree of accuracy ");
 scanf("%lf",&e);

 while(x0<=x)
 {
  printf("x%d = %lf      y%d= %lf\n\n" ,i,x0,i,y0);
  y10 = y0 + h*f(x0,y0);
  while((y10-y0) <= e)
  {
   y11 = y0+(h/2)*(f(x0,y0)+f(x0+h,y0+y10));
   y10 = y11;
  }
  y0 = y10;
  x0 = x0 + h;
  i++;
  delay(500);
 }
 getch();
 }
```

Newton Raphson’s Method:


≡ File Edit Search Run Compile Debug Project Options Window Help
NEWTON.C
8
1:1
Activate Windows
Go to Settings to activate Windo
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

```
double ff(double x)
{
  return(x*x*x-5*x-11);
}
double df(double x)
{
 return(3*x*x - 5);
}



void main()
{
 double x0,e,f,g,x1,x;
 int step = 1;
 clrscr();
 printf("Enter Initial guess ");
 scanf("%lf",&x0);
 printf("Enter degree of accuracy ");
 scanf("%ld",&e);
 clrscr();
```


≡ File Edit Search Run Compile Debug Project Options Window Help
NEWTON.C
8
41:9
Activate Windows
Go to Settings to activate Windo
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

```
 clrscr();
 printf("STEP   X0      f(x0)   f'(x0)  x1\n");
 do
 {
  f = ff(x0);
  g = df(x0);
  if(g == 0)
  {
   printf("Mathematical error ");
   exit(0);
  }
  x1  = x0 - f/g;
  printf("%d\t%lf\t%lf\t%lf\t%lf\n",step,x0,f,g,x1);
  step++;
  x = x0;
  x0 = x1;

  }while(fabs(x0-x)>e);
  printf("Root is %lf",x1);
 getch();
 }
```