INDEX

S. No.	***Date***	***Name of the Program***	***Page No.***	***Marks Awarded***	***Remarks***
1		Newton - Raphson method			
2		Gauss - Seidel method			
3		Lagrange’s Interpolation method			
4		Trapezoidal rule			
5		Runge - Kutta method of fourth order			
6		Adam’s predictor and corrector method			

Program 1: Newton-Raphson method

Question:

Find a positive real root of $x^3 - x - 2 = 0$ using Newton-Raphson method.

Program:

```
def f(x):
    return x**3-x-2
def f1(x):
    return 3*x**2-1
xo=float (input ("Enter the initial approximation: "))
for i in range (1,10):
    xn=xo-f(xo)/f1(xo)
    xo=xn
print ("The approximate root using Newton-Raphson method is %.4f"%xn)
```

Output:

```
Enter the initial approximation: 1
The approximate root using Newton-Raphson method is 1.5214
```

Program 2: Gauss-Seidel method

Question:

Solve the system of equations $4x + y + z = 1$; $x + 3y + z = 2$; $x + y + 5z = 3$, using Gauss-Seidel method.

Program:

```
x0=0; y0=0; z0=0
for i in range (1,10):
    x=1/4*(1-y0-z0)
    x0=x
    y=1/3*(2-x0-z0)
    y0=y
    z=1/5*(3-x0-y0)
    z0=z
print ("The approximate solution of x = %.4f, y= %.4f, z=%.4f"% (x, y,
z))
```

Output:

```
The approximate solution of x = 0.0000, y= 0.5000, z=0.5000
```

Program 3: Lagrange's Interpolation method

Question:

Using Lagrange interpolation formula, find the value corresponding to $x = 10$ from the following table

x	0	1	2	4	5	6
y	1	14	15	5	6	19

Program:

```
x= [0,1,2,4,5,6]
y= [1,14,15,5,6,19]
s=float (input ("Enter the value of x to be in: "))
sum=0
for i in range (0,6):
   prod=1
for j in range (0,6):
   if i!=j:
       prod=prod*(s-x[j])/(x[i]-x[j])
       sum=sum+prod*y[i]
print ("The functional value is %.4f"%sum)
```

Output:

```
Enter the value of x to be in: 10
The functional value is 2254.6667
```

Program 4: Trapezoidal rule

Question:

Evaluate $\int_0^1 \frac{dx}{1+x^2}$ using trapezoidal rule with $h = 0.2$.

Program:

```
def f(x):
    return 1/(1+x**2)
a=float (input ("Enter the lower limit: "))
b=float (input ("Enter the upperlimit: "))
h=float (input ("Enter the step size: "))
n=int((b-a)/h)
sum=0
for i in range (1, n):
     sum=sum+f(a+i*h)
trap=h/2*(f(a)+f(b)+2*sum)
print ("The Integral value is %.5f"%trap)
```

Output:

```
Enter the lower limit: 0
Enter the upperlimit: 1
Enter the step size: 0.2
The Integral value is 0.78373
```

Program 5: Runge-Kutta method of fourth order

Question:

Find $y(0.2)$, given that $\frac{dy}{dx} = x + y^2, y(0) = 1$ using Runge-Kutta fourth order method.

Program:

```
def f (x, y):
   return x+y**2
x0=float (input ("Enter initial point of x: "))
y0=float (input ("Enter initial point of y: "))
h=float (input ("Enter step value h: "))
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)
y=y0+(k1+2*k2+2*k3+k4)/6
print ("The value of y using RK method is %.4f"%y)
```

Output:

```
Enter initial point of x: 0
Enter initial point of y: 1
Enter step value h: 0.2
The value of y using RK method is 1.2735
```

Program 6: Adam's predictor and corrector method

Question:

Evaluate (1.4) , given that $\frac{dy}{dx} = x^2(1 + y), y(1) = 1, y(1.1) = 1.233, y(1.2) = 1.548$ and $y(1.3) = 1.979$ using Adam's predictor and corrector method.

Program:

```
def f (x, y):
      return x**2*(1+y)
x0=float (input ("Enter x0: "))
y0=float (input ("Enter y0: "))
x1=float (input ("Enter x1: "))
y1=float (input ("Enter y1: "))
x2=float (input ("Enter x2: "))
y2=float (input ("Enter y2: "))
x3=float (input ("Enter x3: "))
y3=float (input ("Enter y3: "))
h =0.1
y4p=y3+(h/24) *(55*f (x3, y3)-59*f (x2, y2) +37*f (x1, y1)-9*f (x0,
y0))
x4=x3+h
y4c=y3+(h/24) *(9*f (x4, y4p) +19*f (x3, y3)-5*f (x2, y2) +f (x1, y1))
print ("Approximate soln is %0.4f"%y4c)
```

Output:

```
Enter x0: 1
Enter y0: 1
Enter x1: 1.1
Enter y1: 1.233
Enter x2: 1.2
Enter y2: 1.548
Enter x3: 1.3
Enter y3: 1.979
Approximate solution is 2.5749
```