LAB - 1

Title: Newton Raphson Method for finding root of a function

Objective:

- To use Newton Raphson method to find a root of a function

Source code:

```
#include<stdio.h>
#include<main.h>
float f(float x){
    return (cos(x) + 2*sin(x) + x*x);
}
float df(float x){
    return (-sin(x) + 2*cos(x) + 2*x);
}
void main(){
    float a, tol, b;
    printf("Enter initial guess: ");
    scanf("%f", &a);
    printf("Enter tolerance value: ");
    scanf("%f", &tol);
    do{
        b = a;
        a = b-(f(b)/df(b));
    } while(fabs(a-b)>tol);
    printf("\n The approximate root is %f\n", b);
}
```

Output

```
Enter initial guess: 2
Enter tolerance value: 0.00001
The approximate root is -0.653926.
```

LAB - 2

Title: Bisection Method for finding the root of a function

Objective :- To find the root of a function using Bisection Method

Source Code:

```
#include <stdio.h>
#include <math.h>
float f(x) {
    return pow(x, 3) - x - 2;
}
float bisection (float l, float r) {
    float mid = (l+r) / 2;
    while (fabs (f(mid)) >= 0.000001) {
        if (f(mid) * f(l) < 0) r = mid;
    } else {
        l = mid;
        mid = (l+r) / 2;
    }
    return mid;
}
void main() {
    float root = bisection (1,2);
    printf ("The root of the function is approximately : %.6f\n",
                                                    root);
}
```

Output:

The root of the function is approximately : 1.521980

LAB-9

Title: Fitting a linear line in the given data set

Objective:

- To fit a linear linear line in the given data set

Source Code:-

```
#include<stdio.h>
void main(){
  int n,i;
  float x[10], y[10], sx, sy, sx2, sxy, a_b;
  sx=sy=sxy=sx2=0;
  printf("How many data points; ");
  scanf("%d", &n);
  printf("Enter %d data points for x & y:\n", n);
  for(i=0; i<n; i++){
    scanf("%f %f", &x[i], &y[i]);
}
  for(i=0; i<n; i++){
    sx+=x[i];
    sy+=y[i];
    sxy+=x[i]*y[i];
    sx2+=x[i]*y[i];
  }
  b=((n*sxy)-(sx*sy))/((n*sx2)-(sx*sx));
  a=(sy/n)-(b*sx/n);
  printf("The fitted line is: y=%0.2f+%0.2fx", a_b);
}
```

Output:

```
How many data points: 3
Enter 3 data points for x & y:
1 4
2 7
3 9
```

The fitted line is $y = 1.67 + 2.50x$.

Title: Fitting exponential model to the given data set:

Objective: To fit an exponential model to given data set

Source code:

```
#include<stdio.h>
void main(){
   int n,i;
   float x[10], y[10], sx, slogy, sx2, sxy, a, b, r;
   sx = slogy = sxy = sx2 = 0;
   printf("How many data points: ");
   scanf("%d", &n);
   printf("Enter %d data points for x & y:\n", n);
   for(i=0; i<n; i++){
      scanf("%f %f", &x[i], &y[i]);
   }
   for (i = 0; i<n; i++){
      sx += x[i];
      slogy += log(y[i]);
      sxy += x[i] * log(y[i]);
      sx2 += x[i] * x[i];
   }
   b = ((n * sxy) - (sx * slogy))/((n * sx2) - (sx * sx));
   r = (slogy/n) - (b * sx/n);
   a = (exp(r));
   printf("The fitted exponential curve is: y = %0.2f e^ %0.2fx", a, b);
}
```

Output

How many data points: 3

Enter 3 data points for x & y

1 2

2 5

3 11

The fitted exponential curve is 0.87e^0.85x.

LAB - 5

Title: Interpolation of the given data set using Lagrange's Interpolation

Objective: - To interpolate the given data set using Lagrange's Interpolation

Code:

```
#include<stdio.h>
#include<math.h>
void main(){
  float x[20], y[20], xp, yp=0, p;
  int i, j, n;
  printf("Enter the number of points: ");
  scanf("%d", &n);
  printf("Enter X and Y values : \n");
 for(i=0; i<n; i++){
   scanf("%f %f", &x[i], &y[i]);
}
 printf("\n Enter the value of x for which y is to be found: ");
 scanf("%f", &xp);
for(i=0; i<n; i++){
     p=i;
     for(j=1; j<n; j++){
       if(i!=j) p=p*(xp-x[j])/(x[i]-x[j]);
   }
   yp+=p*y[i];
}
printf("Interpolated value = %f", yp);
}
```

Output:

```
Enter the number of points: 3
Enter X and Y values:
   1  1
   2  4
   3  9
Enter the value of x for which y is to be found: 4
Interpolated value = 16.000000
```

LAB-6

Title: Two point forward formula for derivating a function

Objective :- To find the derivative of a function using Two point forward-Formula

Source Code:

```
#include<stdio.h>
#define F(x) (x*x+2*x+2)
void main() {
  float xi, x1, h, f1, f2, d;
  printf("Enter value of xi and h: ");
  scanf("%f %f", &xi, &h);
  f1 = f(xi);
  x1 = xi+h;
  f2 = f(x1);
  d = (f2-f1)/h;
printf("The derivative of a function at %f = %f\n", xi, d);
}
```

Output:

```
Enter the value of xi and h
   2   0.25
The derivative of a function at 2.000000 = 6.250000.
```

LAB-7

Title: Two point Backward Formula for derivating a function

Objective:

- To find the derivative of a function using Two Point Backward-Formula

Source Code:

```
#include<stdio.h>
#define F(x) (x*x+2*x+1)
void main(){
   float x1, xi, h, f1, f2, d;
   printf("Enter the value of xi and h :\n");
   scanf("%f %f", &xi, &h);
   f1 = F(xi);
   x1 = xi - h;
   f2 = F(x1);
   d = (f1-f2)/h;
   printf("The derivative of a function at %f = %f", xi, d);
}
```

Output:

```
Enter the value of xi and h :
1  0.25
The derivative of a function at 1.000000 = 3.750000.
```

LAB-8

Title: Three Point Formula for Derivative of a function

Objective:

- To find the derivative of a function using Three Point Formula

Source Code:

```
#include<stdio.h>
#define F(x)(x*x+2*x+2)
void main(){
    float xi, x1, x2, h, f1, f2, d;
    printf("Enter the value of xi and h :");
    scanf("%f %f", &xi, &h);
    f2= f(xi);
    x1 = xi+h;
    x2 = xi-h;
    f1 = F(x1);
    f2 = f(x2);
    d= (f1-f2)/(2*h);
    printf("The derivation of a function at %f= %f", xi, d);
}
```

Output:

```
Enter the value of xi and h : 10.25
The derivation of a function at %f = %f", xi, d);
```

LAB-9

Title: Integration of a function using Trapezoidal rule

Objective:

- To integrate a function using Trapezoidal rule

Source Code:

```
#include<stdio.h>
#include<math.h>
#define F(x) (exp(x))
void main() {
    float h, x1, x2, f1, f2, I;
    printf("Enter lower limit & upper limit: ");
    scanf("%f %f", &x1, &x2);
    h = x2 - x1;
    f1 = F(x1);
    f2 = F(x2);
    I = (f1+f2)*h/2;
    printf("Integration by Trapezoidal Rule = %f", I);
}
```

Output:

```
Enter lower limit & upper limit: 1 4
Integration by Trapezoidal Rule = 85.974640
```

LAB-10

Title: Integration of a function using Composite Trapezoidal Rule

Objective:

- To integrate a function using Composite Trapezoidal Rule

Source Code:

```
#include<stdio.h>
#include<math.h>
#define f(x) x*x
void main() {
    int n=100;
    float a=1, b=4;
    float h=(b-a)/n;
    float sum=0;
    for(int i=1; i<n; i++){
        float x=a+i*h;
        sum+=f(x);
    }
    float integral=(h/2)*f(a)+f(b)+2*sum);
    printf("The integral is %f", integral);
}
```

Output

The integral is 21.000450

LAB-11

Title: Implementation of Euler's Method for solving ordinary differential equations

Objective: - To implement Euler's Method for solving ordinary differential equations

Source Code:

```
#include<stdio.h>
#define f(x,y) (x+y)
int main() {
    float x0=0, y0=2, xn=1, h, yn;
    int n=10;
    h=(xn-x0)/n;
    for(int i=0; i<n; i++) {
        yn=y0+h*f(x0,y0);
        y0=yn;
        x0+=h;
    }
    printf("Value of y at x=%0.2f is %.03f", xn, yn);
    return 0;
}
```

Output:

value of y at x=1.00 is 5.1782.

LAB-12

Title: Implementation of Heun's Method for solving Ordinary differential Equations

Objectives:

- To implement Heun's method for solving ordinary differential eq^.

Source Code :

```
#include <stdio.h>
#define f(x,y) (x+y)
int main() {
  float x0=0, y0=1, h=0.1, xn=1, x, y;
  int n=(xn-x0)/h+1;
  for(int i=0; i<n; i++){
      x = x0+i*h;
      y=y0+h*(f(x,y0)+f(x+h,y0+h*f(x,y0)))/2;
      y0=y;
  }
  printf("Value of y at x=%0.2f is %0.2f", x,y);
  return 0;
}
```

Output :

Value of y at x = 1.00 is 3.90.