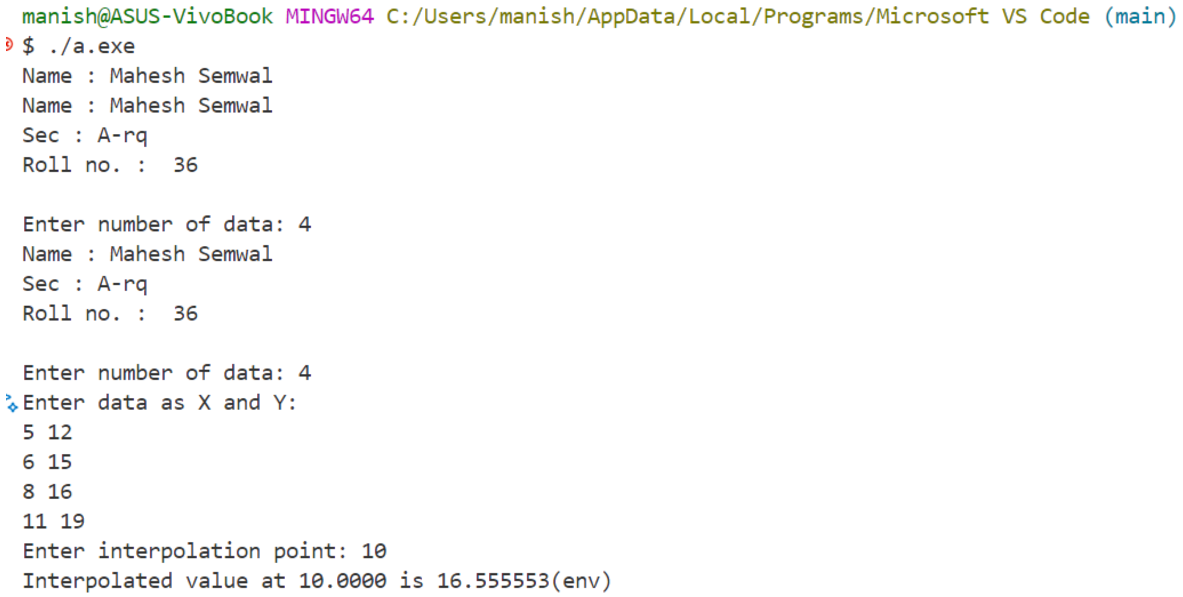
LAGRANGE INTERPOLATION METHOD

Applications:

1. **Data Interpolation**: Find intermediate values between known data points in fields like physics, economics, and engineering.
2. **Numerical Integration**: Used in Newton-Cotes formulas to approximate integrals from discrete data.
3. **Curve Fitting**: Fit curves through given data points to analyze trends or approximate functions.
4. **Computer Graphics**: Generate smooth transitions and animations by interpolating curves.
5. **Error Estimation**: Estimate and analyze errors in numerical approximations.

```c
#include <stdio.h>
void main(){
    printf("Name : Mahesh Semwal\n");
    printf("Sec : A-rq\n");
    printf("Roll no. :  36\n\n");
    float point, Y = 0, p;
    int i, j, n;
    printf("Enter number of data: ");
    scanf("%d", &n);
    float x[n], y[n];
    printf("Enter data as X and Y:\n");
    for (i = 0; i < n; i++)
    {
        scanf("%f%f", &x[i], &y[i]);
    }
    printf("Enter interpolation point: ");
    scanf("%f", &point);
    for (i = 0; i < n; i++){
        p = 1;
        for (j = 0; j < n; j++){
            if (i != j){
```

```
            p = p * (point - x[j]) / (x[i] - x[j]);
          }
        }
        Y = Y + p * y[i];
    }
    printf("Interpolated value at %.4f is %f", point, Y);
}
```



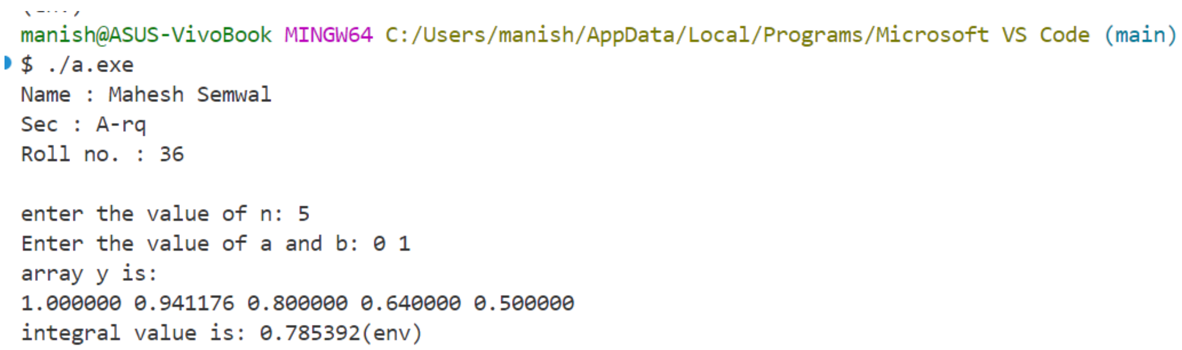
```
manish@ASUS-VivoBook MINGW64 C:/Users/manish/AppData/Local/Programs/Microsoft VS Code (main)
$ ./a.exe
Name : Mahesh Semwal
Name : Mahesh Semwal
Sec : A-rq
Roll no. :  36

Enter number of data: 4
Name : Mahesh Semwal
Sec : A-rq
Roll no. :  36

Enter number of data: 4
Enter data as X and Y:
5 12
6 15
8 16
11 19
Enter interpolation point: 10
Interpolated value at 10.0000 is 16.555553(env)
```

SIMPSON’S ONE-THIRD RULE

Applications:

1. **Physics**: Calculating areas under velocity-time graphs to determine displacement.
2. **Engineering**: Estimating work done by varying forces.
3. **Probability**: Computing probabilities for complex probability density functions.
4. **Economics**: Approximating integrals in cost, revenue, and demand functions.
5. **Astronomy**: Determining distances using integral-based formulas in orbital mechanics.

```
#include<stdio.h>
double fun(double x){
    return 1.0/(1+x*x);
}
int main(){
    printf("Name : Mahesh Semwal\n");
    printf("Sec : A-rq\n");
    printf("Roll no. : 36\n\n");
    int n;
    double a,b;
    printf("enter the value of n: ");
    scanf("%d",&n);
    printf("Enter the value of a and b: ");
    scanf("%lf%lf",&a,&b);
    double h=(b-a)/(n-1);
    double y[n];
    for(int i=0;i<n;i++){
        y[i]=fun(a+i*h);
    }
    double sum=y[0]+y[n-1];
    for(int i=1;i<n-1;i++){
        if(i%2!=0){
            sum=sum+4*y[i];
        }
        else{
            sum=sum+2*y[i];
        }
    }
    printf("array y is: \n");
    for(int i=0;i<n;i++){
        printf("%lf ",y[i]);
    }
    sum=(h*sum)/3;
    printf("\nintegral value is: %lf",sum);

}
```



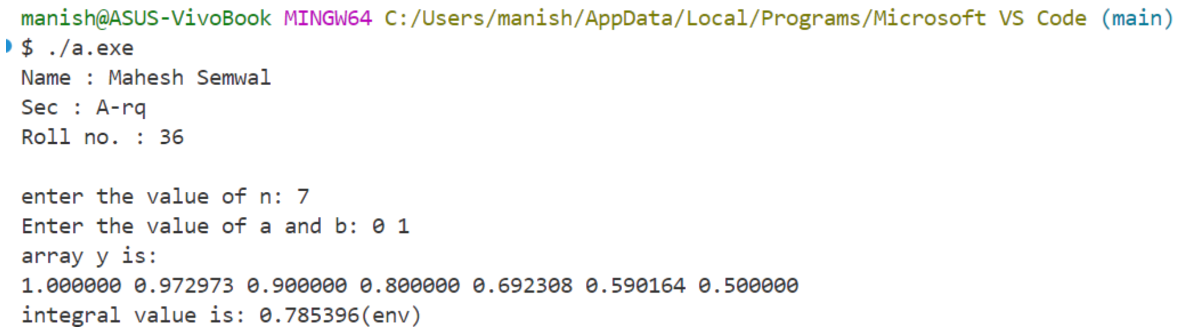
```
manish@ASUS-VivoBook MINGW64 C:/Users/manish/AppData/Local/Programs/Microsoft VS Code (main)
$ ./a.exe
Name : Mahesh Semwal
Sec : A-rq
Roll no. : 36

enter the value of n: 5
Enter the value of a and b: 0 1
array y is:
1.000000 0.941176 0.800000 0.640000 0.500000
integral value is: 0.785392(env)
```

SIMPSON’S THREE-EIGHT RULE

Applications:

1. **Physics**: Accurately calculates work done by forces with cubic or higher-order variations.
2. **Engineering**: Used in systems like fluid dynamics or heat transfer, where the function has non-linear behavior.
3. **Probability**: Computes probabilities for distributions with complex higher-order density functions.
4. **Economics**: Approximates integrals in models with rapidly changing variables, like cost or revenue curves.

```
#include<stdio.h>

double fun(double x){

  return 1.0/(1+x*x);

}


int main(){

  printf("Name : Mahesh Semwal\n");

  printf("Sec : A-rq\n");

  printf("Roll no. : 36\n\n");

  int n;

  double a,b;

  printf("enter the value of n: ");

  scanf("%d",&n);

  printf("Enter the value of a and b: ");

  scanf("%lf%lf",&a,&b);

  double h=(b-a)/(n-1);

  double y[n];

  for(int i=0;i<n;i++){

    y[i]=fun(a+i*h);

  }

  double sum=y[0]+y[n-1];

  for(int i=1;i<n-1;i++){
        if(i%3==0){
            sum=sum+2*y[i];
        }
        else{
            sum=sum+3*y[i];
        }
    }
    printf("array y is: \n");
    for(int i=0;i<n;i++){
        printf("%lf ",y[i]);
    }
    sum=(3.0*h*sum)/8;
    printf("\nintegral value is: %lf",sum);
}
```



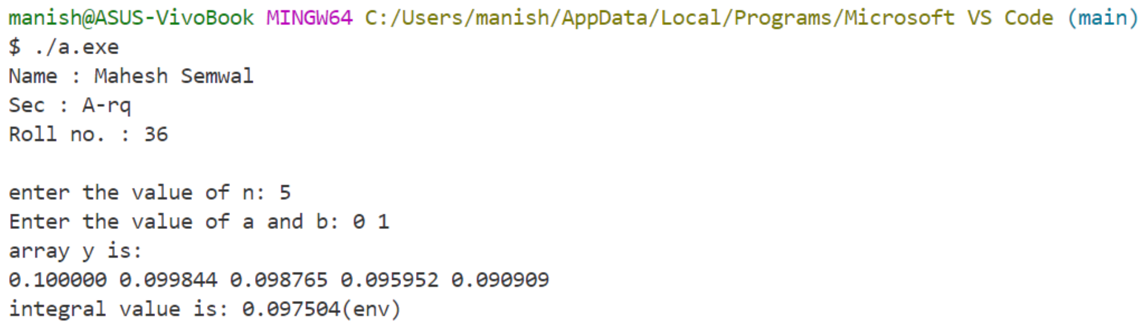
```
manish@ASUS-VivoBook MINGW64 C:/Users/manish/AppData/Local/Programs/Microsoft VS Code (main)
$ ./a.exe
Name : Mahesh Semwal
Sec : A-rq
Roll no. : 36

enter the value of n: 7
Enter the value of a and b: 0 1
array y is:
1.000000 0.972973 0.900000 0.800000 0.692308 0.590164 0.500000
integral value is: 0.785396(env)
```

TRAPEZOIDAL

Applications:

1. **Physics**: Calculating approximate displacement from velocity-time data.
2. **Engineering**: Estimating flow rates or heat transfer in systems with linear variations.
3. **Medicine**: Computing drug concentration over time in pharmacokinetics.
4. **Environmental Science**: Measuring total pollutant levels from concentration-time curves.
5. **Geography**: Approximating areas of irregular land plots.

```
#include<stdio.h>

#include<math.h>

double fun(double x){

  return 1/(10+pow(x,3));

}


int main(){

  printf("Name : Mahesh Semwal\n");

  printf("Sec : A-rq\n");

  printf("Roll no. : 36\n\n");

  int n;

  double a,b;

  printf("enter the value of n: ");

  scanf("%d",&n);

  printf("Enter the value of a and b: ");

  scanf("%lf%lf",&a,&b);

  double h=(b-a)/(n-1);

  double y[n];

  for(int i=0;i<n;i++){

    y[i]=fun(a+i*h);

  }

  double sum=y[0]+y[n-1];
  for(int i=1;i<n-1;i++){
    sum=sum+2*y[i];
  }
  printf("array y is: \n");
  for(int i=0;i<n;i++){
    printf("%lf ",y[i]);
  }
  sum=(h*sum)/2;
  printf("\nintegral value is: %lf",sum);

}
```



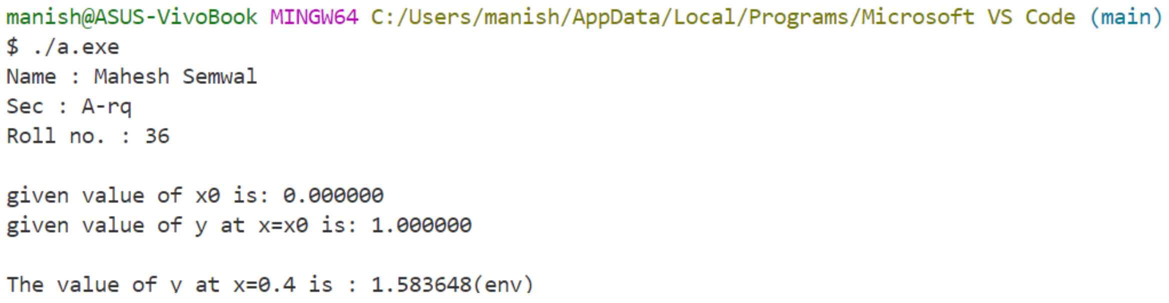
```
manish@ASUS-VivoBook MINGW64 C:/Users/manish/AppData/Local/Programs/Microsoft VS Code (main)
$ ./a.exe
Name : Mahesh Semwal
Sec : A-rq
Roll no. : 36

enter the value of n: 5
Enter the value of a and b: 0 1
array y is:
0.100000 0.099844 0.098765 0.095952 0.090909
integral value is: 0.097504(env)
```

RUNGE-KUTTA METHOD

Applications:

1. **Physics**: Solving equations for motion, oscillations, and wave phenomena (e.g., pendulums, projectile motion).
2. **Engineering**: Modeling dynamic systems like robotics, vehicle motion, and control systems.
3. **Electronics**: Analyzing RLC circuits and transient responses.
4. **Astronomy**: Simulating orbital mechanics and celestial dynamics.
5. **Biology**: Modeling population dynamics and biological growth using differential equations.

```
#include <stdio.h>
float dydx(float x, float y)
{
    return x+y;
}


float rungeKutta(float x0, float y0, float x, float h)
{
    int n = (int)((x - x0) / h);
    float k1, k2, k3, k4, k5;
    float y = y0;
    for (int i = 1; i <= n; i++)
    {
        k1 = h * dydx(x0, y);
        k2 = h * dydx(x0 + 0.5 * h, y + 0.5 * k1);
        k3 = h * dydx(x0 + 0.5 * h, y + 0.5 * k2);
        k4 = h * dydx(x0 + h, y + k3);
        y = y + (1.0 / 6.0) * (k1 + 2 * k2 + 2 * k3 + k4);
        x0 = x0 + h;
    }
    return y;
}

int main()
{
    printf("Name : Mahesh Semwal\n");
    printf("Sec : A-rq\n");
    printf("Roll no. : 36\n\n");
    float x0 = 0, y = 1, x = 0.4, h = 0.1;
    printf("given value of x0 is: %f\n",x0);
    printf("given value of y at x=x0 is: %f\n",y);
    printf("\nThe value of y at x=0.4 is : %f",rungeKutta(x0, y, x, h));
    return 0;
}
```



```
manish@ASUS-VivoBook MINGW64 C:/Users/manish/AppData/Local/Programs/Microsoft VS Code (main)
$ ./a.exe
Name : Mahesh Semwal
Sec : A-rq
Roll no. : 36

given value of x0 is: 0.000000
given value of y at x=x0 is: 1.000000

The value of y at x=0.4 is : 1.583648(env)
```