

## 1.Fibonacci series up to a given no. of terms

```
#include<stdio.h>
int main(){
    int n,i;
    double a,b,c;
    printf("Enter the number of terms: ");
    scanf("%d",&n);
    if (n<0){
        printf("You can't enter negative value of n");
        return 0;
    }

    a=0;
    b=1;
    printf("%.0lf, ",a);
    for(i=1;i<n;i++){
        c = a+b;
        a=b;
        b=c;
        printf("%.0lf, ",a);
    }
    return 0;
}
```

### Output :

```
Enter the number of terms: 15
0, 1 1 2 3 5 8 13 21 34 55 89 144 233 377
```

## 2.Fibonacci series upto a given sum.

```
#include<stdio.h>
int main(){
    int a = 0, b = 1, sum = 0, c = 0, n;
    printf("Enter the sum upto which Fibonacci series is to be printed: ");
}
```

```
scanf("%d",&n);
while(b<=n){
    printf("%d\t",a);
    sum = a+b;
    a = b;
    b = sum;
    c += a;}
}
```

## Output :

```
Enter the sum upto which Fibonacci series is to be printed: 250
0    1    1    2    3    5    8    13   21   34   55   89   144
```

## 3.Sin Series

```
#include<stdio.h>
#include <math.h>
int main(){
int n;
    float sum,x,term;
    printf("Enter value of x(in degrees) and n: ");
    scanf("%f %d",&x,&n);
    x=x*3.14/180;
    sum = x;
    term = x;
    for(int i=1;i<n;i++){
        term = ((-1)*pow(x,2)*term)/(2*i*(2*i+1));
        sum += term;
    }
    printf("%f",sum);
}
```

## Output :

```
Enter value of x(in degrees) and n: 45 100
0.706825
```

## 4.Value of polynomial.

```
#include<stdio.h>
int main(){
    int n,i,j;
    float y,x;
    printf("Enter the order of polynomial:");
    scanf("%d",&n);
    if (n<0){
        printf("You can't enter negative value of n !!");
        return 0;
    }
    float arr[n+1];
    for(i=n;i>=0;i--){
        printf("Enter the coefficient of a[%d]: ",i);
        scanf("%f",&arr[i]);
    }
    printf("Enter value of x: ");
    scanf("%f",&x);
    y=arr[n];
    for(i=n;i>0;i--){
        y=arr[i-1]+x*y;
    }

    printf("Value of Polynomial : %f\n",y);
    return 0;
}
```

## Output :

```
Enter the order of polynomial:2
Enter the coefficient of a[2]: 1
Enter the coefficient of a[1]: 1
Enter the coefficient of a[0]: 1
Enter value of x: 4
Value of Polynomial : 21.000000
```

## 5.Polynomial Division

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

```
int main(){
    int i, n;
    double r;

    printf("Enter the value of r : ");
    scanf("%lf",&r);
    printf("Enter the order of polynomial : ");
    scanf("%d",&n);
    if (n<0){
        printf("You can't enter negative value of n !!");
        return 0;
    }

    float a[n+1];
    float b[n];
    for(i=n;i>=0;i--){
        printf("Enter the coefficient of a[%d]: ",i);
        scanf("%f",&a[i]);
    }

    b[n-1] = a[n];
    for (i=1; i<=n-1;i++){
        b[n-i-1] = a[n-i] + r*b[n-i];
    }

    for(i=0;i<=n-1;i++){
        printf("\nCoefficient of x^%i : %.2f",i,b[i]);
    }
    return 0;
}
```

## Output :

```
Enter the value of r : 2
Enter the order of polynomial : 2
Enter the coefficient of a[2]: 1
Enter the coefficient of a[1]: -4
Enter the coefficient of a[0]: 4

Coefficient of x^0 : -2.00
Coefficient of x^1 : 1.00
```

## 6. Binary to decimal

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<string.h>
int main(){
    int j,n,sum=0;
    char binary[20];
    printf("Enter the Binary Number :");
    gets(binary);
    n=strlen(binary);
    for(j=0;j<n;j++){
        if((binary[j]!='1')&&(binary[j]!='0')){
            printf("\nPlease enter a Binary number !");
            exit(1);
        }
        sum+=(binary[n-1-j]-48)*(pow(2,j));
    }
    printf("Decimal equivalent of the given Binary Number : %d\n",sum);
}
```

## Output :

```
Enter the Binary Number :1101
Decimal equivalent of the given Binary Number : 13
```

## 7. Decimal to binary

```
#include<stdio.h>
int main(){
    int bin[100];
    int n,i,j,a,m=0;
    printf("Enter the decimal number :");
    scanf("%d",&n);
    for(i=n,j=0;i>0;i/=2,j++){
        a=i%2;
        bin[j]=a;
        m++;
    }
    printf("binary equivalent is : ");
    for(i=m-1;i>=0;i--){
```

```

    printf("%d",bin[i]);
}
return 0;
}

```

## Output :

```

Enter the decimal number :23
binary equivalent is : 10111

```

## 8.Bisection method

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

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

```
int sign(float x,float y);
```

```
float eval(float *arr,int n,float x);
```

```
int main(){
```

```
    int n,i=0,j;
```

```
    float x,lower_bound,upper_bound;
```

```
    printf("Enter the order of polynomial:");
```

```
    scanf("%d",&n);
```

```
    if (n<0){
```

```
        printf("You can't enter negative value of n !!");
```

```
        return 0;
```

```
    }
```

```
    float arr[n+1];
```

```
    for(i=n;i>=0;i--){
```

```
        printf("Enter the coefficient of a[%d]: ",i);
```

```
        scanf("%f",&arr[i]);
```

```
    }
```

```
    printf("Enter lower bound : ");
```

```
    scanf("%f",&lower_bound);
```

```
    printf("Enter upper bound : ");
```

```
    scanf("%f",&upper_bound);
```

```
for(x=lower_bound;x<=upper_bound;x=x+1){
printf("value of y(%.2f) : %.2f\n",x,eval(arr,n,x));}
```

```
float x0,x1,e,x2;
printf("Enter the value of x0 ,x1 ,e: ");
scanf("%f %f %f",&x0,&x1,&e);
```

```
float y0,y1,y2;
y0=arr[n];
y1=arr[n];
for(i=n-1;i>=0;i--)
{
    y0=arr[i]+x0*y0;
    y1=arr[i]+x1*y1;
}
```

```
if (sign(y0,y1)){
    printf("Starting values are unsuitable\n");
    printf("Values of x0,x1,y0,y1,e: %f %f %f %f
%f\n",x0,x1,y0,y1,e);
    return 0;
}
```

```
while(fabs((x1-x0)/x1)>e){
```

```
    x2=(x1+x0)/2;
    y2=arr[n];
    for(j=n-1;j>=0;j--){
        y2=arr[j]+x2*y2;
    }
```

```
i=i+1;
```

```
//replacing values of x0 or x1 to x2 based on sign of y0 and y1
```

```
if (sign(y0,y2)){
    x0=x2;
    y0=y2;
}
```

```
else{
```

```
        x1=x2;
        y1=y2;
    }
}

printf("No. of iterations : %d\n",i);
printf("Value of root : %f\n",x2);

return 0;
}

int sign (float x,float y){
    if (x*y>0)
        return 1;
    else if(x*y<0)
        return 0;
}

float eval(float *arr,int n,float x){
    float y;
    y=arr[n];
    for(int i=n;i>0;i--){
        y=arr[i-1]+x*y;
    }
    return y;
}
```

**Output :**



```
Enter the order of polynomial:2
Enter the coefficient of a[2]: 1
Enter the coefficient of a[1]: 0
Enter the coefficient of a[0]: -16
Enter lower bound : 1
Enter upper bound : 10
value of y(1.00) : -15.00
value of y(2.00) : -12.00
value of y(3.00) : -7.00
value of y(4.00) : 0.00
value of y(5.00) : 9.00
value of y(6.00) : 20.00
value of y(7.00) : 33.00
value of y(8.00) : 48.00
value of y(9.00) : 65.00
value of y(10.00) : 84.00
Enter the value of x0 ,x1 ,e: 2 7 0.001
No. of iterations : 10
Value of root : 3.999512
```

## 9.Regula falsi method.

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

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

```
int sign(float x,float y);
```

```
float eval(float *arr,int n,float x);
```

```
int main(){
```

```
    int n,i=0,j,m;
```

```
    float x,lower_bound,upper_bound;
```

```
    printf("Enter the order of polynomial:");
```

```
    scanf("%d",&n);
```

```
    if (n<0){
```

```
        printf("You can't enter negative value of n !!");
```

```
        return 0;
```

```
    }
```

```
    float arr[n+1];
```

```
    for(i=n;i>=0;i--){
```

```

    printf("Enter the coefficient of a[%d]: ",i);
    scanf("%f",&arr[i]);
}
printf("Enter lower bound : ");
scanf("%f",&lower_bound);
printf("Enter upper bound : ");
scanf("%f",&upper_bound);

for(x=lower_bound;x<=upper_bound;x=x+1){
    printf("value of y(%.2f) : %.2f\n",x,eval(arr,n,x));
}
float x0,x1,e,x2;
printf("Enter the value of x0 ,x1 ,e ,m: ");
scanf("%f %f %f %d",&x0,&x1,&e,&m);

float y0,y1,y2;
y0=arr[n];
y1=arr[n];
for(i=n-1;i>=0;i--)
{
    y0=arr[i]+x0*y0;
    y1=arr[i]+x1*y1;
}

else if (sign(y0,y1))
{
    printf("Starting values are unsuitable\n");
    printf("Values of x0,x1,y0,y1,e: %f %f %f %f\n",x0,x1,y0,y1,e);
    return 0;
}

for(i=0;i<m;i++){
    x2=(y1*x0 - y0*x1)/(y1-y0);
    y2=arr[n];
    for(j=n-1;j>=0;j--){

```

```

        y2=arr[j]+x2*y2;
    }
    if (fabs(y2)<=e){
        printf("Convergent solution x=%f , y=%f\n",x2,y2);
        return 0;
    }

```

**//replacing values of x0 or x1 to x2 based on sign of y0,y1 and y2**

```

    if (sign(y0,y2)){
        x0=x2;
        y0=y2;}
    else{
        x1=x2;
        y1=y2;
    }
}

```

```

printf("Does not converge in %d iterations \n",m);

```

```

    return 0;
}

```

**//function which returns 1 if sign of two arguments is same else returns 0**

```

int sign (float x,float y){
    if (x*y>0)
        return 1;
    else if(x*y<0)
        return 0;
}

```

```

float eval(float *arr,int n,float x){
    float y;
    y=arr[n];
    for(int i=n;i>0;i--)
    {

```

```

    y=arr[i-1]+x*y;
}
return y;
}

```

## Output:-

```

Enter the order of polynomial:2
Enter the coefficient of a[2]: 1
Enter the coefficient of a[1]: 0
Enter the coefficient of a[0]: -25
Enter lower bound : 1
Enter upper bound : 10
value of y(1.00) : -24.00
value of y(2.00) : -21.00
value of y(3.00) : -16.00
value of y(4.00) : -9.00
value of y(5.00) : 0.00
value of y(6.00) : 11.00
value of y(7.00) : 24.00
value of y(8.00) : 39.00
value of y(9.00) : 56.00
value of y(10.00) : 75.00
Enter the value of x0 ,x1 ,e ,m: 3 7 0.001 10
Convergent solution x=4.999947 , y=-0.000534

```

## 10.Newton Raphson Method.

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

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

```
float eval(float *arr,int n,float x);
```

```
int main(){
```

```
    int n,i,m;
```

```
    float x0,x1,e,d,y0,y1,y_0;
```

```
    printf("Enter order of polynomial : ");
```

```
    scanf("%d",&m);
```

```
    float arr[m+1];
```

```
    for (int j = m; j >= 0; j--){
```

```

    printf("Enter coefficient of a[%d] : ",j);
    scanf("%f",&arr[j]);
}

float arr1[m];
for (int j = m-1; j >= 0; j--){
    printf("Enter coefficient of a'[%d] : ",j);
    scanf("%f",&arr1[j]);
}

printf("Enter x0, e, no. of iteration, delta :");
scanf("%f %f %d %f",&x0,&e,&n,&d);

for(int i = 1; i<n; i++){
    y0 = eval(arr, m, x0);
    y_0 = eval(arr1, m-1, x0);

    if (fabs(y_0) == d){
        printf("Slope too small \n x0 = %f , f0 = %f , f'0 = %f ,
iteration = %d",x0,y0,y_0,i);
        return 0;
    }
    x1 = x0 - (y0)/(y_0);
    y1 = eval(arr, m, x1);
    if (fabs((x1-x0)/x1)<=e) {
        printf("Converges to root x = %f , f(x) = %f , num of iteration
= %d",x1,y1,i);
        return 0;
    }
    x0 = x1;
}

printf("Does not converge in %d iterations\n x0 = %f , x1 = %f , f0
= %f , f'0 = %f",n,x0,x1,y0,y_0);
return 0;
}

```

```

float eval(float *arr,int n,float x){
    float y;
    y=arr[n];
    for(int i=n;i>0;i--)
    {
        y=arr[i-1]+x*y;
    }
    return y;
}

```

## Output :

```

Enter order of polynomial : 2
Enter coefficient of a[2] : 1
Enter coefficient of a[1] : 0
Enter coefficient of a[0] : -25
Enter coefficient of a'[1] : 2
Enter coefficient of a'[0] : 0
Enter x0, e, no. of iteration, delta : 7 0.001 10 0.1
Converges to root x = 5.000000 , f(x) = 0.000000 , num of iteration = 4

```

## 11. Gauss Elimination with Pivotal condensation.

```

#include<stdio.h>
#include<stdlib.h>
#include<math.h>
int main()
{
    int i,j,q,k,p,n,m;
    float u,sum,x[100],arr[100][100],max,e,temp;
    printf("Enter the order of system and error e: ");
    scanf("%d%f",&n,&e);
    for(i=0;i<n;i++){
        for(j=0;j<=n;j++){

```

```

printf("Enter the value of arr[%d][%d]: ",i,j);
scanf("%f",&arr[i][j]);
}
}
for(k=0;k<n;k++){
    max=fabs(arr[k][k]);
    p=k;
    for(m=k+1;m<n;m++){
        if(fabs(arr[m][k])>max){
            max=fabs(arr[m][k]);
            p=m;}
    }
    if(max<=e){
        printf("Ill conditioned equation");
        exit(1);}
    else if(p=k)
        goto out;
    else{
        for(q=k;q<=n;q++){
            temp=arr[k][q];
            arr[k][q]=arr[p][q];
            arr[p][q]=temp;
        }
        goto out;
    }
    out:
    for(k=0;k<n-1;k++){
        for(i=k+1;i<n;i++){
            u=arr[i][k]/arr[k][k];
            for(j=k;j<=n;j++){
                arr[i][j]=arr[i][j]-u*arr[k][j];
            }
        }
    }
    x[n-1]=arr[n-1][n]/arr[n-1][n-1];
    for(i=n-2;i>=0;i--){

```

```

        sum=0;
        for(j=i+1;j<n;j++){
            sum+=arr[i][j]*x[j];
        }
        x[i]=(arr[i][n]-sum)/arr[i][i];
    }
    for(i=0;i<n;i++){
        printf("Value of x[%d]= %f\n",i+1,x[i]);
    }
}

```

## Output :

```

Enter the order of system and error e: 2 0.001
Enter the value of arr[0][0]: 1
Enter the value of arr[0][1]: 1
Enter the value of arr[0][2]: 2
Enter the value of arr[1][0]: 2
Enter the value of arr[1][1]: 3
Enter the value of arr[1][2]: 5
Value of x[1]= 1.000000
Value of x[2]= 1.000000

```

## 12.Gauss Siedel Method

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

```

float relErr(float num1, float num2){
    float diff = num1-num2>=0 ? num1-num2 : num2-num1;
    return (diff/num2);
}

```

```

int main(){
    int n,maxltr;
    float err;
    printf("Enter the order of equations : ");
    scanf("%d",&n);

```



```
float coeff[n][n+1];
for(int i=0;i<n;i++){
    for(int j=0;j<n+1;j++){
        printf("Enter the coefficient of arr[%d][%d] : ",i,j);
        scanf("%f",&coeff[i][j]);
    }
}

printf("Enter the value of err and max iterations: ");
scanf("%f %d",&err, &maxltr);
float x[n];
for(int i=0;i<n;i++) x[i]=0;

for(int k=1;k<=maxltr;k++){
    float big = 0;
    for(int i=1;i<=n;i++){
        float sum=0;
        for(int j=1;j<=n;j++){
            if(i!=j){
                sum = sum + (coeff[i-1][j-1]*x[j-1]);
            }
        }
        float temp = (coeff[i-1][n+1-1]-sum)/coeff[i-1][i-1];
        if(relErr(x[i-1], temp)>big){
            big = relErr(x[i-1], temp);
        }
        x[i-1] = temp;
    }
    if(big<=err){
        printf("\nConverges to solution: ");
        for(int i=1;i<=n;i++){
            printf("x(%d)=%f ",i,x[i-1]);
        }
        int dump = 0;
        scanf("%d", dump); //To hold the output on the screen
    }
}
```

```

    return 0; //Equivalent to stopping the program
}
}
printf("Does not converge in given iterations");
printf("Following are the corresponding values of X's obtained: ");
for(int i=1;i<=n;i++){
    printf("x(%d)=%f ",i,x[i-1]);
}
return 0;
}

```

## Output:-

```

Enter the order of equations : 2
Enter the coefficient of arr[0][0] : 1
Enter the coefficient of arr[0][1] : 1
Enter the coefficient of arr[0][2] : 2
Enter the coefficient of arr[1][0] : 2
Enter the coefficient of arr[1][1] : 3
Enter the coefficient of arr[1][2] : 5
Enter the value of err and max iterations: 0.001 20

Converges to solution: x(1)=1.001522 x(2)=0.998985

```

## 13.Lagrange Interpolation.

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

```

int main() {
    int n;
    float x;
    printf("Please enter the value of x(at which value is to be
interpolated) and n: ");
    scanf("%f %d",&x, &n);
    float xVal[n];
    float funcVal[n];

    for(int i=0;i<n;i++){
        printf("Enter value of x[%d] and f[%d]\t",i+1,i+1);

```

```

    scanf("%f %f", &xVal[i], &funcVal[i]);
}

float sum = 0;
for(int i=0;i<n;i++){
    float prodFunc = 1;
    for(int j=0;j<n;j++){
        if(i!=j){
            prodFunc = prodFunc*((x-xVal[j])/(xVal[i]-xVal[j]));
        }
    }
    sum = sum + (funcVal[i]*prodFunc);
}
printf("\nThe value of the function is :%f at x=%f",sum, x);
return 0;
}

```

## Output :

```

Please enter the value of x(at which value is to be interpolated) and n: 2.5 4
Enter value of x[1] and f[1]      1 1
Enter value of x[2] and f[2]      2 8
Enter value of x[3] and f[3]      3 27
Enter value of x[4] and f[4]      4 64

The value of the function is :15.625000 at x=2.500000

```

## 14.Trapezoidal Rule.

```

#include<stdio.h>
int main(){
    int i,j,n;
    float h,sum,integral;
    printf("Enter the value of h :");
    scanf("%f",&h);

    if(h>1){
        printf("Value is not suitable");
        return 0;
    }
}

```

```

printf("Enter no. of terms: ");
scanf("%d",&n);
float f[n+1];
for(i=1;i<n+1;i++){
    printf("Enter the value of f[%d] : ",i);
    scanf("%f",&f[i]);
    sum =(f[1]+f[n])/2;
}
for(j=2;j<n;j++){
    sum += f[j];
}
integral = h * sum;
printf("Value of integral = %f \n",integral);
return 0;
}

```

## Output :

```

Enter the value of h :0.001
Enter no. of terms: 5
Enter the value of f[1] : 0.2474
Enter the value of f[2] : 0.2571
Enter the value of f[3] : 0.2667
Enter the value of f[4] : 0.2764
Enter the value of f[5] : 0.2860
Value of integral = 0.001067

```

## 15.Trapezoidal two point form

```

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

```

```
float eval(float *arr,int n,float x);
```

```
int main(){
```

```

    float x,x1,x2,e,h,s1,s2,s4,l0,l1;
    printf("Enter the value of x1,x2,e: ");
    scanf("%f %f %f",&x1,&x2,&e);

```

```

int n,i;
printf("Enter order of polynomial: ");
scanf("%d",&n);
if (n<0)
{
    printf("You can't enter negative value of n !!");
    return 0;
}
float arr[n+1];
for(int i=n;i>=0;i--)
{
    printf("enter the coefficient of a[%d]: ",i);
    scanf("%f",&arr[i]);
}

```

```

h = (x2-x1)/2;
i = 2;
s1 = (eval(arr,n,x1) + eval(arr,n,x2))/2;
l1 = h * s1;
while (fabs((l1 - l0)/l1) > e){
    x = x1 + h/2;
    for (int j = 0; j < i; j++){
        s1 = s1 + eval(arr,n,x);
        x += h;
    }
    i *= 2;
    h /= 2;

    l0 = l1;
    l1 = h*s1;
}

```

```

printf("Integrated value using trapeziodal 2 point form rule is %f ,
h = %f , i = %d\n",l1,h,i);

```

```

    return 0;
}

float eval(float *arr,int n,float x){
    float y;
    y=arr[n];
    for(int i=n;i>0;i--){
        y=arr[i-1]+x*y;
    }
    return y;
}

```

## Output:-

```

Enter the value of x1,x2,e: 1 2 0.001
Enter order of polynomial: 2
enter the coefficient of a[2]: 3
enter the coefficient of a[1]: 2
enter the coefficient of a[0]: 1
Integrated value using trapeziodal 2 point form rule is 10.989504 , h = 0.000977 , i = 1024

```

## 16.Trapezoidal with refinement.

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

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

```
float eval(float *arr,int n,float x);
```

```

int main(){
    int m,k;
    float x,x1,h,e,s1,l0,l1;
    printf("Enter the values of x1,h,k,e: ");
    scanf("%f %f %d %f",&x1,&h,&k,&e);
    printf("Enter order of polynomial : ");
    scanf("%d",&m);
}

```

```
float arr[m+1];
for (int j = m; j >= 0; j--){
    printf("Enter coefficient of a[%d] : ",j);
    scanf("%f",&arr[j]);
}
x = x1;
float arr1[k+1];
for (int i = 1; i <= k; i++){
    arr1[i] = eval(arr,m,x);
    x += h;
}
s1 = (arr1[1] + arr1[k]) / 2;
for (int j = 2; j <= k-1; j++){
    s1 += arr1[j];
}
I1 = h * s1;
int i = k-1;

while (fabs((I1 - I0)/I1) > e){
    I0 = I1;
    x = x1 + h/2;
    for (int j = 1; j <= i; j++){
        s1 = s1 + eval(arr,m,x);
        x += h;
    }
    i *= 2;
    h /= 2;
}
I1 = s1 * h;
printf("Integration is %f",I1);
return 0;
}

float eval(float *arr,int n,float x){
    float y;
    y=arr[n];
```

```

    for(int i=n;i>0;i--){
        y=arr[i-1]+x*y;
    }
    return y;
}

```

## Output:-

```

Enter the values of x1,h,k,e: 1 0.1 11 0.001
Enter order of polynomial : 2
Enter coefficient of a[2] : 3
Enter coefficient of a[1] : 2
Enter coefficient of a[0] : 1
Integration is 11.001250

```

## 17.Simpsons Rule

```

#include<stdio.h>
int main(){
    int i,n,j,k;
    float h,sum,integral;
    printf("Enter the value of h : ");
    scanf("%f",&h);
    if(h>1){
        printf("h is not suitable");
        return 0;
    }

    printf("Enter the no. of terms:");
    scanf("%d",&n);
    float f[n+1];
    for(i=1;i<n+1;i++){
        printf("Enter the value of a[%d] : ",i);
        scanf("%f",&f[i]);
    }
    sum = (f[1]+f[n]);
    for(j=2;j<n;j+=2){
        sum=sum+4*f[j];
    }
}

```



```

}
for(k=3;k<n-1;k+=2){
    sum=sum+2*f[k];
}
integral = (h*sum)/3;
printf("Value of integral = %f",integral);

return 0;
}

```

## Output:-

```

Enter the value of h : 0.001
Enter the no. of terms:5
Enter the value of a[1] : 0.2474
Enter the value of a[2] : 0.2571
Enter the value of a[3] : 0.2667
Enter the value of a[4] : 0.2764
Enter the value of a[5] : 0.2860
Value of integral = 0.001067

```

## 18.Simpson rule with two point.

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

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

```
float eval(float *arr,int n,float x);
```

```
int main(){
```

```
    int n;
```

```
    float x,x1,x2,e,h;
```

```
    printf("Enter x1, x2, e: ");
```

```
    scanf("%f %f %f",&x1,&x2,&e);
```

```
    h=(x2-x1)/2;
```

```
    printf("Enter order of polynomial: ");
```

```
    scanf("%d",&n);
```

```

float f[n+1];
for(int i=n;i>=0;i--){
    printf("enter the value of a[%d]: ",i);
    scanf("%f",&f[i]);
}

```

```

int i = 2;
float s1 = eval(f,n,x1) + eval(f,n,x2);
float s2 = 0;
float s4 = eval(f,n,x1+h);
float l0 = 0;
float ln = (s1 + 4*s2) * (h/3);

```

```

while(fabs((ln - l0)/ln)>=e){
    s2 = s2+s4;
    s4 = 0;
    x = x1 + h/2;
    for (int j = 1; j < i; j++){
        s4 = s4 + eval(f,n,x);
        x = x + h;
    }
    h = h/2;
    i = 2*i;
    l0 = ln;
    ln = (s1 + 2*s2 + 4*s4) * (h/3);
}
printf("Integration using simpson two point form is %f \n",ln);
printf("Values of h and i are : %f %d\n",h,i);
return 0;
}

```

```

float eval(float *arr,int n,float x){
    float y;
    y=arr[n];
    for(int i=n;i>0;i--){
        y=arr[i-1]+x*y;
    }
}

```

```

    }
    return y;
}

```

## Output:-

```

Enter x1, x2, e: 1 2 0.001
Enter order of polynomial: 2
enter the value of a[2]: 3
enter the value of a[1]: 2
enter the value of a[0]: 1
Integration using simpson two point form is 10.990589
Values of h and i are : 0.000061 16384

```

## 19.Gauss Legendre 5 point Formula.

```

#include<stdio.h>
#include<math.h>
int main(){
    int i;
    float X[50],R[50],l,a,b,F[50],p,q;
    printf("Enter lower and upper limit : ");
    scanf("%f%f",&a,&b);
    for(i=1;i<5;i++){
        printf("Enter R[%d] and X[%d] : ",i,i);
        scanf("%f%f",&R[i],&X[i]);
    }
    p=(a+b)/2;
    q=(b-a)/2;
    l=0;
    for(i=1;i<5;i++){
        F[i]= sin(p+q*X[i]);
        l=R[i]*F[i]+ l;
    }
    l=q*l;
    printf("\n%f",l);
}

```

## Output :

```
Enter lower and upper limit : 0.25 0.29
Enter R[1] and X[1] : 0.347854 -0.8611363
Enter R[2] and X[2] : 0.6521451 -0.3399810
Enter R[3] and X[3] : 0.6521451 0.3399810
Enter R[4] and X[4] : 0.347854 0.8611363

0.010669
```

## 20.Heun's Method

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

```
float function(float x,float y);
```

```
int main(){
    float x1,y1,h,xFinal;
    printf("Enter the initial values of x and y: ");
    scanf("%f %f",&x1,&y1);
    printf("Enter the final value of x: ");
    scanf("%f", &xFinal);
    printf("Enter the value of h [Step increment]: ");
    scanf("%f", &h);
    while(x1<=xFinal){
        printf("\nThe values are X=%f and y=%f", x1, y1);
        float s1 = function(x1, y1);
        float x2 = x1+h;
        float y2 = y1 + (h*s1);
        float s2 = function(x2, y2);
        y2 = y1 + h*((s1+s2)/2);
        x1 = x2;
        y1 = y2;
    }
    return 0;
}
```

```
float function(float x,float y){
    return -1*(x*y);
}
```

## Output :

```
Enter the initial values of x and y: 0 1
Enter the final value of x: 0.25
Enter the value of h [Step increment]: 0.05
```

```
The values are X=0.000000 and y=1.000000
The values are X=0.050000 and y=0.998750
The values are X=0.100000 and y=0.995011
The values are X=0.150000 and y=0.988811
The values are X=0.200000 and y=0.980196
The values are X=0.250000 and y=0.969230
```

## 21.Polygon Method.

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

```
float function(float x,float y);
```

```
int main(){
    float x1,y1,h,xFinal;
    printf("Enter the initial values of x and y: ");
    scanf("%f %f",&x1,&y1);
    printf("Enter the final value of x: ");
    scanf("%f", &xFinal);
    printf("Enter the value of h: ");
    scanf("%f", &h);

    while(x1<=xFinal){
        printf("At x=%6.5f y=%6.5f\n",x1,y1);
        y1 = y1 + (h*function(x1+(h/2), y1 + (h/2)*function(x1,y1)));
        x1+=h;
    }
    return 0;
}
```

```
float function(float x,float y){  
    return -1*(x*y);  
}
```

## Output :

```
Enter the initial values of x and y: 0 1  
Enter the final value of x: 0.25  
Enter the value of h: 0.05  
At x=0.00000 y=1.00000  
At x=0.05000 y=0.99875  
At x=0.10000 y=0.99501  
At x=0.15000 y=0.98881  
At x=0.20000 y=0.98019  
At x=0.25000 y=0.96921
```

## 22.Predictor Corrector method.

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

```
float function(float x,float y){  
    return -1*(x*y);  
}
```

```
int main(){  
    float x1, y1, h, xFinal;  
    printf("Enter the value of x1 and y1 : ");  
    scanf("%f %f",&x1, &y1);  
    printf("Enter the value of h [Step Value] : ");  
    scanf("%f", &h);  
    printf("Enter the final value of x : ");  
    scanf("%f",&xFinal);  
  
    printf("At x=%f y=%f\n",x1,y1);  
    float s1 = function(x1,y1);  
    float x2 = x1+h;  
    float y2 = y1 + h*s1;
```

```
float s2 = function(x2,y2);
y2 = y1 + (h/2)*(s1+s2);

float y3P,y3C,x3,s3;
while(x2<=xFinal){
    printf("At x=%f y=%f\n", x2,y2);
    y3P = y1 + 2*h*s2;
    x3 = x2+h;
    s3 = function(x3, y3P);
    y3C = y1 +(h*(s2+s3))/2;
    y1 = y2;
    y2 = y3C;
    x2 = x3;
}
return 0;
}
```

## Output:-

```
Enter the value of x1 and y1 : 0 1
Enter the value of h [Step Value] : 0.05
Enter the final value of x : 0.25
At x=0.000000 y=1.000000
At x=0.050000 y=0.998750
At x=0.100000 y=0.996262
At x=0.150000 y=0.993773
At x=0.200000 y=0.990056
At x=0.250000 y=0.986344
```