

\*\*\*\*\*

## Polynomial interpolation using Lagrange's Method

\*\*\*\*\*

```
#include <stdio.h>
#include <math.h>
#include <conio.h>
#define MAX 15

int main()
{
    int n, i, j;
    float x[MAX], f[MAX], fp, lf, sum, xp;
    char q;
    printf("\nInput the number of data pairs: ");
    scanf("%d", &n);
    printf("\nInput data pairs x(i) and values f(i) (one set in each line): ");
    for (i=0; i<n; i++)
        scanf("%f %f", &x[i], &f[i]);
    do{
        printf("\nInput x at which interpolation is required: ");
        scanf("%f", &xp);
        sum=0.0;
        for(i=0; i<n; i++){
            lf=1.0;
            for(j=0; j<n; j++){
                if(i!=j)
                    lf=lf*(xp-x[j])/(x[i]-x[j]);
            }
            sum=sum+lf*f[i];
        }
        fp=sum;
        printf("\n Interpolated function value at x=%f is %f.", xp, fp);
        printf("\n Do you want to input another value? (y/n):");
        scanf(" %c", &q);
    }
    while (q=='y');
    getch();
    return 0;
}
```

\*\*\*\*\*

## Polynomial interpolation using Newton's Interpolation

\*\*\*\*\*/

```
#include <stdio.h>
#include <math.h>
#include <conio.h>
#define MAX 15

int main()
{
    int i, j, n;
    float xp, fp, sum, pi, x[MAX], f[MAX], a[MAX], d[MAX][MAX];
    char q;
    printf("\nInput the number of data pairs: ");
    scanf("%d", &n);
    printf("\nInput data pairs x(i) and values f(i) (one set in each line): ");
    for(i=1; i<=n; i++)
        scanf("%f %f", &x[i], &f[i]);

    /*Construct difference table*/
    for (i=1; i<=n; i++)
        d[i][1]=f[i];
    for (j=2; j<=n; j++)
        for (i=1; i<=n-j+1; i++)
            d[i][j]=(d[i+1][j-1]-d[i][j-1])/(x[i+j-1]-x[i]);

    /*Set the coefficients of interpolation polynomial*/
    for(j=1; j<=n; j++)
        a[j]=d[1][j];

    /*Compute interpolation value*/
    do{
        printf("\nInput x at which interpolation is required:");
        scanf("%f", &xp);
        sum=a[1];
        for(i=2; i<=n; i++){
            pi=1.0;
            for(j=1; j<=i-1; j++)
                pi=pi*(xp-x[j]);
            sum=sum+a[i]*pi;
        }
        fp=sum;

        /*write results*/
        printf("\n Interpolated function value at x=%f is %f.", xp, fp);
        printf("\n Do you want to input another value? (y/n):");
        scanf(" %c", &q);
    }
    while (q=='y');
    getch();
    return 0;
}
```

```

/*****
Newton's Interpolation using Forward Difference Formula
*****/

```

```

#include<stdio.h>
#include<conio.h>
#include<math.h>
#define MAX 15

int main()
{
    float x[MAX], fx[MAX], fd[MAX], xp, s, h, v, p;
    int n, i, j;
    char q;
    printf("\n Input the number of data pairs: ");
    scanf("%d",&n);
    printf("\n Input data pairs x(i) and values f(i) (one set in each line): ");
    for(i=1;i<=n;i++)
        scanf("%f%f", &x[i], &fx[i]);
    h=x[2]-x[1];
    do {
        printf("\n Enter the value at which interpolation is required: ");
        scanf("%f", &xp);
        s=(xp-x[1])/h;
        p=1;
        v=fx[1];
        for(i=1;i<=n;i++)
            fd[i]=fx[i];
        for(i=1;i<=n-1;i++) {
            for(j=1;j<=n-i;j++) {
                fd[j]=fd[j+1]-fd[j];
            }
            p=p*(s-i+1)/i;
            v=v+p*fd[1];
        }
        printf("\n Interpolated function value at x=%f is %f.", xp, v);
        printf("\n Do you want to input another value? (y/n): ");
        scanf(" %c", &q);
    } while (q=='y');
    getch();
    return 0;
}

```

\*\*\*\*\*

## Newton's Interpolation using Backward Difference Formula

\*\*\*\*\*

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
#define MAX 15

int main()
{
    float x[MAX], fx[MAX], bd[MAX], xp, s, h, v, p;
    int n, i, j, k;
    char q;
    printf("\n Input the number of data pairs: ");
    scanf("%d",&n);
    printf("\n Input data pairs x(i) and values f(i) (one set in each line): ");
    for(i=1;i<=n;i++)
        scanf("%f%f", &x[i], &fx[i]);
    h=x[2]-x[1];
    do {
        printf("\n Enter the value at which interpolation is required: ");
        scanf("%f", &xp);
        s=(xp-x[n])/h;
        p=1;
        v=fx[n];
        for(i=1;i<=n;i++)
            bd[i]=fx[i];
        for(i=n, k=1;i>=1, k<n;i--, k++) {
            for(j=n;j>=n-i;j--) {
                bd[j]=bd[j]-bd[j-1];
            }
            p=p*(s+k-1)/k;
            v=v+p*bd[n];
        }
        printf("\n Interpolated function value at x=%f is %f.", xp, v);
        printf("\n Do you want to input another value? (y/n): ");
        scanf(" %c", &q);
    } while (q=='y');
    getch();
    return 0;
}
```

/\*\*\*\*\*\*

## Fitting a linear equation

\*\*\*\*\*

```
#include <stdio.h>
#include <math.h>
#include <conio.h>
#define MAX 10

int main()
{
    int i, n;
    float x[MAX], y[MAX];
    float sumx=0.0, sumy=0.0, sumxx=0.0, sumxy=0.0, xmean, ymean, denom, a, b;

    /*Reading data values*/
    printf("\n Input the number of data points: ");
    scanf("%d", &n);
    printf("\n Input x and y values (one set on each line): ");
    for(i=1; i<=n; i++)
        scanf("%f%f", &x[i], &y[i]);

    /*Computing constants a and b*/
    for(i=1; i<=n; i++){
        sumx=sumx+x[i];
        sumy=sumy+y[i];
        sumxx=sumxx+x[i]*x[i];
        sumxy=sumxy+x[i]*y[i];
    }
    xmean=sumx/n;
    ymean=sumy/n;
    denom=n*sumxx-sumx*sumx;
    b=(n*sumxy-sumx*sumy)/denom;
    a=ymean-b*xmean;

    printf("\n The line that is fit to the given data is y = %f + %fx.", a, b);
    getch();
    return 0;
}
```

\*\*\*\*\*

## Fitting a polynomial equation

\*\*\*\*\*

```
#include <stdio.h>
#include <math.h>
#include <conio.h>
#define MAX 10

void normal (float x[MAX], float y[MAX], float c[MAX][MAX], float b[MAX], int n, int m){
    int i, j, k, l1, l2;
    for(j=1;j<=m;j++){
        for(k=1;k<=m;k++){
            c[j][k]=0.0;
            l1=k+j-2;
            for (i=1;i<=n;i++)
                c[j][k]=c[j][k]+pow(x[i], l1);
        }
    }
    for(j=1;j<=m;j++){
        b[j]=0.0;
        l2=j-1;
        for(i=1;i<=n;i++)
            b[j]=b[j]+y[i]*pow(x[i], l2);
    }
    return;
}

void gauss(int n, float a[MAX][MAX], float b[MAX], float x[MAX]){
    int i, j, k;
    float pivot, factor, sum;
    for(k=1;k<=n-1;k++){
        pivot=a[k][k];
        for(i=k+1;i<=n;i++){
            factor=a[i][k]/pivot;
            for(j=k+1;j<=n;j++)
                a[i][j]=a[i][j]-factor*a[k][j];
            b[i]=b[i]-factor*b[k];
        }
    }
    x[n]=b[n]/a[n][n];
    for(k=n-1;k>=1;k--){
        sum=0.0;
        for(j=k+1;j<=n;j++)
            sum=sum+a[k][j]*x[j];
        x[k]=(b[k]-sum)/a[k][k];
    }
    return;
}
```

```

int main()
{
    int n, mp, m, i;
    float x[MAX], y[MAX], c[MAX][MAX], a[MAX], b[MAX];

    /*Reading Values*/
    printf("\n Input number of data points: ");
    scanf("%d", &n);
    printf("\n Input degree of required polynomial: ");
    scanf("%d", &mp);
    printf("\n Input x and y values (one set on each line): ");
    for(i=1;i<=n;i++)
        scanf("%f%f", &x[i], &y[i]);

    /*Number of polynomial coefficients*/
    m=mp+1;

    /*Computation of elements of c and b*/
    normal(x, y, c, b, n, m);

    /*Computation of coefficients a(1) to a(m)*/
    gauss(m, c, b, a);

    /*Printing of coefficients a(i)*/
    printf("\n Polynomial Coefficients \n\n");
    for(i=1;i<=m;i++)
        printf("%15.6f", a[i]);
    getch();
    return 0;
}

```

```

/*****
Fitting an exponential equation
*****/

```

```

#include <stdio.h>
#include <math.h>
#include <conio.h>
#define MAX 10

```

```

int main()
{
    int i, n;
    float x[MAX], y[MAX], u[MAX];
    float sumx=0.0, sumu=0.0, sumxx=0.0, sumxu=0.0, xmean, umean, denom, a, b;

```

```

/*Reading data values*/
printf("\n Input number of data points: ");
scanf("%d", &n);
printf("\n Input x and y values (one set on each line): ");
for(i=1;i<=n;i++)
    scanf("%f%f", &x[i], &y[i]);
for(i=1;i<=n;i++)
    u[i]=log(y[i]);

```

```

/*Computing constants a and b*/
for(i=1;i<=n;i++){
    sumx=sumx+x[i];
    sumu=sumu+u[i];
    sumxx=sumxx+x[i]*x[i];
    sumxu=sumxu+x[i]*u[i];
}
xmean=sumx/n;
umean=sumu/n;
denom=n*sumxx-sumx*sumx;
b=(n*sumxu-sumx*sumu)/denom;
a=exp(umean-b*xmean);

```

```

printf("\n The exponential equation that is fit to the given data is y = %f e^%fx.", a, b);
getch();
return 0;
}

```