```
#include<bits/stdc++.h>

#include<math.h>

using namespace std;


double mat[100][100], a[100], R;


void polynomialRegression(int n);

void gaussJordan(int n);

void table (double a[]);

double* XY(double x[], double y[],int power, int n);

double correlation(double x[], double y[], int n);

double summation(double arr[], int power, int n);


int main(){


    double x[100], f[100];

    int n;


    printf("Enter the number of data points: ");

    scanf("%d", &n);


    printf("Enter the values of X: ");


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


        scanf("%lf", &x[i]);

    }
```

```
printf("\n");

printf("Enter the values of f(x): ");

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

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

printf("\n");

for(int i = 0; i < 4; i++){

    for(int j = 0; j < 5; j++){

        if(i == 0 && j == 0){

            mat[i][j] = n;
        }

        else if (j != 4){

            mat[i][j] = summation(x, i+j, n);

        } else if(j == 4){
            mat[i][j] = summation(XY(x, f, i, n), 1, n);
        }
    }
}

gaussJordan(4);
polynomialRegression(5);

if(correlation(x, f, n) > 0){

    printf("\n\nStrong positive relation.\n");

}

else if(correlation(x, f, n) < 0){

    printf("\n\nStrong negative relation.\n");
}

else{

    printf("\n\nNo relationship at all.\n");
}

table(a);

return 0;
```

```
}

double summation(double arr[], int power, int n){

    double result = 0, res[100];

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

        res[i] = pow(arr[i], power);
    }

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

        result += res[i];
    }

    return result;
}

double* XY(double x[], double y[],int power, int n){

   static double xy[100], powX[100];

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

        powX[i] = pow(x[i], power);
    }
    for(int i = 0; i < n; i++){

        xy[i] = powX[i] * y[i];
    }

    return xy;
}

void polynomialRegression(int n){

    printf("\nThe polynomial is : ");

    for(int i = 0; i < n - 1; i++){

        if(i == 0){

            printf("%0.4lf ", a[i]);
        }

        else{

            if(a[i] < 0){

                printf("- %0.4lf X ^ %d ", fabs(a[i]), i);
            }
            else{

                printf("+ %0.4lf X ^ %d ", a[i], i);
            }

        }
    }

    printf("\n\n");

    printf("Function\t\t\t\tValue\n");

    for(int i=0; i<n-1; i++){
        printf("A%d\t\t\t\t %lf\n",i+1, a[i]);
    }

}

void gaussJordan(int n){

    int r = 0;
    double norm = 0, x = 0;

    while(r < n){

        norm = 1/mat[r][r];
        //normalize
        for(int i = 0; i <= n; i++){

            mat[r][i] = mat[r][i] * norm;

        }
        //removing x
        for(int i = 0; i < n; i++){

            x = mat[i][r];

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

                if(i != r){

                    mat[i][j] -= (x*mat[r][j]);
                }

                else{

                    break;
                }
            }
        }
        r++;
    }
```

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

        a[i] = mat[i][n]/mat[i][i];
    }
}

double correlation(double x[], double y[], int n){

    double R;

    R = ((n * summation(XY(x, y, 1, n), 1, n)) - (summation(x, 1, n) * summation(y, 1, n))) /
(sqrt(((n * summation(x, 2, n)) - pow(summation(x, 1, n), 2)) * ((n * summation(y, 2, n)) -
pow(summation(y, 1, n), 2))));

    printf("Correlation Coefficient R\t %lf",R);
    return R;
}

void table (double a[]){

    printf("\nYear\t\tAverage Temperature\n");

    for(double x = 2020; x <= 2040; x+=1){

        printf("%0.2lf\t\t%lf\n", x,(a[0] + (a[1] * x) + (a[2] * pow(x, 2)) + (a[3] * pow(x, 3))));
    }
}
```

/*

15

1999

2000

2001

2002

2003

2004

2005

2006

2007

2008

2009

2010

2011

2012

2013

25.71483

25.15017

25.337

25.38033

25.28083

```
25.38633

25.532

25.76567

25.34375

25.3895

25.90492

25.94033

25.20508

25.53358

25.9675


*/
```