/\*

Newton's backward Interpolation Method

\*/

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include <float.h>

long fact(long n) {

if (n == 0)

return 1;

else

return n \* fact(n - 1);

}

int main() {

int n;

printf("Enter the number of values availiable: ");

scanf("%d", &n);

double arr[n][2];

printf("Enter the values [x y]:\n");

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

scanf("%lf %lf", &arr[i][0], &arr[i][1]);

}

/\* generating difference table \*/

double delta[n][n];

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

delta[i][0] = arr[i][1];

}

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

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

delta[j][i] = delta[j + 1][i - 1] - delta[j][i - 1];

}

}

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

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

printf("%lf\t|", delta[i][j]);

}

printf("\n");

}

/\* taking input for the value \*/

double x;

printf("Enter the value of x: ");

scanf("%lf", &x);

double h = arr[1][0] - arr[0][0];

double u = (x - arr[n-1][0]) / h;

if(fabs(u) > 1){

printf("Warning: the value of u is > 1.\n");

}

/\* newton backword interpolation \*/

double y = 0;

printf("\n");

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

double mul = 1;

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

mul \*= (u - j);

}

y += (mul \* delta[n-i-1][i]) / fact(i);

printf(" %d approximation = %lf\n", i, y);

}

printf("\n");

printf("\* h = %lf\n", h);

printf("\* u = %lf\n", u);

printf("The approximate value of y: %lf ", y);

return 0;

}