

ІНТЕРПОЛЯЦІЯ ДАНИХ.
ІНТЕРПОЛЯЦІЙНИЙ ПОЛІНОМ ЛАГРАНЖА.
ЧИСЕЛЬНЕ ДИФЕРЕНЦІЮВАННЯ.

Алгоритми

Лабораторна робота №1

Варіант 7

Виконав: Конча Вадим

Завдання:

1. Запрограмуйте обчислення інтерполяційного поліному $L(x)$ довільного степеню.
2. Функція $f(x)$ задана аналітично та у вигляді таблиці. Згідно з вашим варіантом:
 - а) побудуйте таблицю значень функції $f(x)$ на відрізку $[a, b]$ з кроком h
 - отриманих безпосередньо за формулою;
 - отриманих за допомогою глобальної інтерполяції поліномом Лагранжа.
 - б) обчисліть наближене значення функції $f(x)$ у вказаній точці $x_* \in [a, b]$ за допомогою кускової інтерполяції поліномами Лагранжа вказаного степеню (значення x_* та вузлів інтерполяції, необхідних для побудови полінома, введіть з клавіатури).
3. Побудуйте таблицю значень функцій $f'(x)$ та $f''(x)$ на відрізку $[a, b]$ з кроком h :
 - а) отриманих безпосереднім диференціюванням;
 - б) отриманих за формулами числового диференціювання. Для обчислення значень функції, що використовуються у формулах диференціювання, скористайтесь кусково-поліноміальною інтерполяцією вказаного степеню.

Умова:

7. $f(x) = \exp(x) \sin(x^3)$;

x	2.51	2.62	2,7	2,81	2.91	2.959	3	3.01	3.04
f(x)	-1.2932	-10,4527	11.0147	-3.2493	-8.6484	13.4931	19.2093	17.1088	3.74
x	3.05	3.07	3.08						
f(x)	-2.0721	-13.2096	-17.6195						

$[a, b] = [2.5, 3]$; $h = 0,01$; кубічна інтерполяція.

Код:

//1-е НЕ работает адекватно polinya, хз почему

```
#include <iostream>
#include <vector>
#include <iomanip>    // std::setprecision
#include <cmath>     //exp i sin
```

```
using namespace std;
```

```
vector<float> x = { 2.51, 2.62, 2.7, 2.81, 2.91, 2.959, 3.0, 3.01, 3.04, 3.05, 3.07, 3.08 };
```

```
vector<float> y = {  
    -1.2932, -10.4527, 11.0147, -3.2493, -8.6484, 13.4931, 19.2093, 17.1088, 3.74,  
    -2.0721, -13.2096, -17.6195  
};
```

```
vector<float> xsavearr = x, ysavearr = y;
```

```
vector<float> t;
```

```
vector<float> xnew, ynew; //донорские массивы с нужными нам точками  
интерполейшина
```

```
float a = 2.5, asave = a, b = 3, h = 0.01;
```

```
void genT() {  
    for (a = 2.5; a <= b; a += h) {  
        t.push_back(a);  
    }  
    a = asave;  
}
```

```
//polinya находит значение х-а благодаря данным х и у
```

```
float polinya(vector<float> x, vector<float> y, float tt) {
```

```
    float polinomial = 0; // переменная под сумму полиномов  
    float polinomstarter=0;  
    for (int j = 0; j < y.size(); j++) { //мб будут проблемы с нулём, попробуй сменить  
на 1 j //+ один фор для (y) и грика  
        float chisl = 1, znam = 1, l; //единицы потому что умножаем же, камон)  
        for (int i = 0; i < x.size(); i++) { //второй фор для (x) икса  
            if (i != j) { //для знам'а, щоб на ноль не делить, а то будет грусть  
                chisl *= (tt - x[i]);  
                znam *= (x[j] - x[i]); //обрати внимание что тут есть *=, так что это не  
один слогаемое  
                l = chisl / znam; //увесь той страшный доданок  
            }  
        }  
        polinomstarter = l * y[j]; //одна из многих частей полинома, которые потом  
будут суммой  
        polinomial += polinomstarter;  
    }  
    return polinomial;
```

```

}

float funcmywe(float x) {
    return exp(x) * sin(pow(x, 3));
}

void generate_dots(vector<float> x, vector<float> y, float x_custom)
{//поиск точек с минимальной разницей
    for (int k = 0; k < 4; k++) //повтор 4 раза, для поиска ЧЕТЫРЁХ ТОЧЕК
    {
        float min = x[0];
        int indexfor_y = 0;
        for (int i = 0; i < x.size()-1; i++)
            //перебор элементов для создания min
        {
            if (abs(x_custom - x[i]) > abs(x_custom - x[i + 1])) //проверка разницы
между элементами с нужным числом
            {
                min = x[i + 1]; //новый минимум
                indexfor_y = i + 1;
            }
        }
        xnew.push_back(min); //иксы есть
        ynew.push_back(y[indexfor_y]);
        //cout << "x= " << min << " y=" << y[indexfor_y] << endl; //проверка
        x.erase(x.begin() + indexfor_y);
        y.erase(y.begin() + indexfor_y);
    }
}

float diff(float x_cust)
{
    float fdef;
    x = xsavearr;
    y = ysavearr;
    //polinya находит значение x-а благодаря данным x и y
    fdef = (polinya(x, y, x_cust + h) - polinya(x, y, x_cust)) / h;
    return fdef;
}

float ddiff(float x_cust)
{
    float fdef;
    x = xsavearr;
    y = ysavearr;
    //polinya находит значение x-а благодаря данным x и y
    fdef = (diff(x_cust+h) - diff(x_cust)) / h;
}

```

```

    return fdef;
}
float diffreal(float x)
{
    return 3 * pow(x, 2) * pow(exp(1), x) * cos(pow(x, 3)) + pow(exp(1), x) * sin(pow(x,
3));
}
float ddiffreal(float x)
{
    return -9 * pow(x, 4) * pow(exp(1), x) * sin(pow(x, 3)) + 6 * pow(x, 2) * pow(exp(1),
x) * cos(pow(x, 3)) + 6 * x * pow(exp(1), x) *
    cos(pow(x, 3)) + pow(exp(1), x) * sin(pow(x, 3));
}

int main()
{
    genT(); //ф-я что формулирует область значений

    /*
        проверка ГенТА
    int i = 0;
    while (true) {

        i++;
        cout << t[i] << endl;
    }
    cout << "The end.";
    */
    //cout.precision(10);
    for (int k = 0; k < t.size(); k++) {
        cout << k+1 << " " << "x = " << t[k] << "\t   that is = " << polinya(x, y, t[k]) << "\t
like this = " << funcmywe(t[k]) << endl;
    }

    //////////// В
    float x_custom; //просит x, для кубической интерполяции
    cout << "\n" << "Enter your x in [a;b]: ";
    cin >> x_custom;
    //проверка на границы
    if (a-h>= x_custom || x_custom>=b+h || !cin)
    {
        cout << "Input n0t [a;b] ERROR" << " a: " << a << " b: " << b;
        exit(0);
    }
}

```

//кубическая будет заключаться в том, что мы найдём нужные точки,
//составим из них массивы, и будем юзать просто "полиня" уже без лишних значений

//итак для куба нам надо 4 точки интерполяции

```
generate_dots(x, y, x_custom);
```

```
cout << "\nDOTS ARE:" << endl;
```

```
for (int i = 0; i < xnew.size(); i++)  
{  
    cout << "\n" << xnew[i] << "    --- xnew elem\t" << ynew[i] << "    --- ynew elem";  
}
```

```
    cout << "\n\nx = " << x_custom << "\t    that is = " << polinya(xnew, ynew,  
x_custom)  
    << "\t    like this = " << funcmywe(x_custom) << endl;
```

//////////////////////////////////// C

```
vector <float> diff_real_first;  
for (int k = 0; k < t.size(); k++) {  
    float diffur = diffreal(t[k]);  
    diff_real_first.push_back(diffur);  
}  
vector <float> diff_real_second;  
for (int k = 0; k < t.size(); k++) {  
    float diffur = ddiffreal(t[k]);  
    diff_real_second.push_back(diffur);  
}
```

//для формулы числового диф-а - нужно использовать кусочную интерполяцию для поиска $x-a + h$


```
vector<float> diff_arr_first;  
for (int k = 0; k < t.size(); k++) {  
    float diffur = diff(t[k]);  
    diff_arr_first.push_back(diffur);  
}  
vector<float> diff_arr_second;  
for (int k = 0; k < t.size(); k++) {  
    float diffur = ddiff(t[k]);  
    diff_arr_second.push_back(diffur);  
}  
cout << setprecision(3);
```

```

    for (int k = 0; k < diff_real_first.size(); k++) {
        cout << "\n\n" << k + 1 << ") " << t[k] << endl
            << "RealF: \t" << diff_real_first[k] << "\t"
            << "LagrF: \t" << diff_arr_first[k] << "\t"
            << "RealS: \t" << diff_real_second[k] << "\t"
            << "LagrS: \t" << diff_arr_second[k] << "\t";
    }
}

```

Вивід:

 Консоль отладки Microsoft Visual Studio

1) x = 2.5	that is = 107.656	like this = 1.00954
2) x = 2.51	that is = -1.2932	like this = -1.29316
3) x = 2.52	that is = -64.0781	like this = -3.61402
4) x = 2.53	that is = -94.7969	like this = -5.86713
5) x = 2.54	that is = -103.688	like this = -7.96401
6) x = 2.55	that is = -99.0781	like this = -9.81695
7) x = 2.56	that is = -86.6875	like this = -11.3427
8) x = 2.57	that is = -71.5859	like this = -12.4661
9) x = 2.58	that is = -55.8359	like this = -13.1241
10) x = 2.59	that is = -41.2656	like this = -13.2694
11) x = 2.6	that is = -28.625	like this = -12.8733
12) x = 2.61	that is = -18.3369	like this = -11.9289
13) x = 2.62	that is = -10.4527	like this = -10.4527
14) x = 2.63	that is = -4.54785	like this = -8.48553
15) x = 2.64	that is = -0.246094	like this = -6.09292
16) x = 2.65	that is = 2.8584	like this = -3.36354
17) x = 2.66	that is = 5.12793	like this = -0.406863
18) x = 2.67	that is = 6.92285	like this = 2.65056
19) x = 2.68	that is = 8.4292	like this = 5.66993
20) x = 2.69	that is = 9.79211	like this = 8.50617
21) x = 2.7	that is = 11.0147	like this = 11.0147
22) x = 2.71	that is = 12.0741	like this = 13.0587
23) x = 2.72	that is = 12.8723	like this = 14.517
24) x = 2.73	that is = 13.3159	like this = 15.291
25) x = 2.74	that is = 13.2903	like this = 15.312
26) x = 2.75	that is = 12.6785	like this = 14.5469
27) x = 2.76	that is = 11.4219	like this = 13.0024
28) x = 2.77	that is = 9.50623	like this = 10.7279
29) x = 2.78	that is = 6.97314	like this = 7.81546
30) x = 2.79	that is = 3.8988	like this = 4.3978

```
Консоль отладки Microsoft Visual Studio
25) x = 2.74      that is = 13.2903      like this = 15.312
26) x = 2.75      that is = 12.6785      like this = 14.5469
27) x = 2.76      that is = 11.4219      like this = 13.0024
28) x = 2.77      that is = 9.50623      like this = 10.7279
29) x = 2.78      that is = 6.97314      like this = 7.81546
30) x = 2.79      that is = 3.8988       like this = 4.3978
31) x = 2.8       that is = 0.430969      like this = 0.64373
32) x = 2.81      that is = -3.24921      like this = -3.24916
33) x = 2.82      that is = -6.92595      like this = -7.06404
34) x = 2.83      that is = -10.3661      like this = -10.5763
35) x = 2.84      that is = -13.3397      like this = -13.5669
36) x = 2.85      that is = -15.6261      like this = -15.8354
37) x = 2.86      that is = -17.0457      like this = -17.2148
38) x = 2.87      that is = -17.4613      like this = -17.5835
39) x = 2.88      that is = -16.7996      like this = -16.8765
40) x = 2.89      that is = -15.0536      like this = -15.0938
41) x = 2.9       that is = -12.2901      like this = -12.3049
42) x = 2.91      that is = -8.64859      like this = -8.64858
43) x = 2.92      that is = -4.33547      like this = -4.32917
44) x = 2.93      that is = 0.386284      like this = 0.393349
45) x = 2.94      that is = 5.21204       like this = 5.21717
46) x = 2.95      that is = 9.81423       like this = 9.81687
47) x = 2.96      that is = 13.864        like this = 13.8648
48) x = 2.97      that is = 17.0544       like this = 17.0543
49) x = 2.98      that is = 19.1239       like this = 19.1237
50) x = 2.99      that is = 19.8783       like this = 19.8782
51) x = 3         that is = 19.2094       like this = 19.2094

Enter your x in [a;b]: 2.86
```

Видно що Поліном не погано працює на проміжку з [2.7;3], але на проміжку [2.5;2.69] результати показують дуже сильні розбіжності, зводячись лише у деяких точках. З чим і виходять ще більші розбіжності у наступних завданнях, де апроксимація грає ключову роль. Не розумію чому так виходить, алгоритм перевірявся, та навіть були спроби перевірити його несправність іншим кодом з інтернет-ресурсів. Результати ті ж.

Функція на перший погляд повинна працювати правильно, але так не стається. Можливо не дуже вдало була підібрана саме функція з умови, або її вузли, що і спричинило такі розбіжності у результатах. Якщо є думки, з чим це може бути пов'язано, дуже радий буду отримати відповідь.

Друга та третя частини завдання:

```
Консоль отладки Microsoft Visual Studio

Enter your x in [a;b]: 2.86

DOTS ARE:

2.81 --- xnew elem    -3.2493 --- ynew elem
2.91 --- xnew elem    -8.6484 --- ynew elem
2.959 --- xnew elem   13.4931 --- ynew elem
3 --- xnew elem       19.2093 --- ynew elem

x = 2.86          that is = -23.3812    like this = -17.2148

1) 2.5
RealF:  -227    LagrF:  -1.09e+04    RealS:  -991    LagrS:  4.62e+05
2) 2.51
RealF:  -233    LagrF:  -6.28e+03    RealS:  -186    LagrS:  3.21e+05
3) 2.52
RealF:  -230    LagrF:  -3.07e+03    RealS:   675    LagrS:  2.18e+05
4) 2.53
RealF:  -219    LagrF:   -889    RealS:  1.56e+03    LagrS:  1.35e+05
5) 2.54
RealF:  -199    LagrF:   461    RealS:  2.44e+03    LagrS:  7.78e+04
6) 2.55
RealF:  -170    LagrF:  1.24e+03    RealS:  3.28e+03    LagrS:  2.71e+04
```

```
Консоль отладки Microsoft Visual Studio

5) 2.54
RealF:  -199    LagrF:   461    RealS:  2.44e+03    LagrS:  7.78e+04
6) 2.55
RealF:  -170    LagrF:  1.24e+03    RealS:  3.28e+03    LagrS:  2.71e+04
7) 2.56
RealF:  -134    LagrF:  1.51e+03    RealS:  4.03e+03    LagrS:  6.48e+03
8) 2.57
RealF:   -90    LagrF:  1.58e+03    RealS:  4.67e+03    LagrS: -1.18e+04
9) 2.58
RealF: -40.8    LagrF:  1.46e+03    RealS:  5.14e+03    LagrS: -1.93e+04
10) 2.59
RealF:  12.2    LagrF:  1.26e+03    RealS:  5.43e+03    LagrS: -2.35e+04
11) 2.6
RealF:  67.1    LagrF:  1.03e+03    RealS:  5.5e+03    LagrS: -2.4e+04
12) 2.61
RealF:  122    LagrF:   788    RealS:  5.34e+03    LagrS: -1.98e+04
13) 2.62
RealF:  173    LagrF:   590    RealS:  4.93e+03    LagrS: -1.6e+04
14) 2.63
RealF:  219    LagrF:   430    RealS:  4.27e+03    LagrS: -1.2e+04
```

```
Консоль отладки Microsoft Visual Studio

13) 2.62
RealF: 173   LagrF: 590   Reals: 4.93e+03   LagrS: -1.6e+04

14) 2.63
RealF: 219   LagrF: 430   Reals: 4.27e+03   LagrS: -1.2e+04

15) 2.64
RealF: 258   LagrF: 310   Reals: 3.39e+03   LagrS: -8.35e+03

16) 2.65
RealF: 286   LagrF: 227   Reals: 2.29e+03   LagrS: -4.75e+03

17) 2.66
RealF: 303   LagrF: 179   Reals: 1.02e+03   LagrS: -2.89e+03

18) 2.67
RealF: 306   LagrF: 151   Reals: -375   LagrS: -1.43e+03

19) 2.68
RealF: 295   LagrF: 136   Reals: -1.83e+03   LagrS: -1.4e+03

20) 2.69
RealF: 270   LagrF: 122   Reals: -3.28e+03   LagrS: -1.63e+03

21) 2.7
RealF: 230   LagrF: 106   Reals: -4.66e+03   LagrS: -2.61e+03

22) 2.71
RealF: 177   LagrF: 79.8   Reals: -5.88e+03   LagrS: -3.55e+03
```

```
Консоль отладки Microsoft Visual Studio

21) 2.7
RealF: 230   LagrF: 106   Reals: -4.66e+03   LagrS: -2.61e+03

22) 2.71
RealF: 177   LagrF: 79.8   Reals: -5.88e+03   LagrS: -3.55e+03

23) 2.72
RealF: 113   LagrF: 44.4   Reals: -6.87e+03   LagrS: -4.69e+03

24) 2.73
RealF: 40.6   LagrF: -2.56   Reals: -7.56e+03   LagrS: -5.86e+03

25) 2.74
RealF: -37   LagrF: -61.2   Reals: -7.89e+03   LagrS: -6.45e+03

26) 2.75
RealF: -116   LagrF: -126   Reals: -7.83e+03   LagrS: -6.59e+03

27) 2.76
RealF: -192   LagrF: -192   Reals: -7.34e+03   LagrS: -6.17e+03

28) 2.77
RealF: -261   LagrF: -253   Reals: -6.41e+03   LagrS: -5.41e+03

29) 2.78
RealF: -319   LagrF: -307   Reals: -5.08e+03   LagrS: -3.93e+03

30) 2.79
RealF: -362   LagrF: -347   Reals: -3.39e+03   LagrS: -2.12e+03
```

```
Консоль отладки Microsoft Visual Studio

30) 2.79
RealF: -362 LagrF: -347 RealS: -3.39e+03 LagrS: -2.12e+03

31) 2.8
RealF: -386 LagrF: -368 RealS: -1.4e+03 LagrS: 34.4

32) 2.81
RealF: -389 LagrF: -368 RealS: 774 LagrS: 2.37e+03

33) 2.82
RealF: -370 LagrF: -344 RealS: 3.03e+03 LagrS: 4.67e+03

34) 2.83
RealF: -329 LagrF: -297 RealS: 5.23e+03 LagrS: 6.87e+03

35) 2.84
RealF: -266 LagrF: -229 RealS: 7.25e+03 LagrS: 8.67e+03

36) 2.85
RealF: -185 LagrF: -142 RealS: 8.93e+03 LagrS: 1e+04

37) 2.86
RealF: -89 LagrF: -41.6 RealS: 1.02e+04 LagrS: 1.08e+04

38) 2.87
RealF: 16.4 LagrF: 66.2 RealS: 1.08e+04 LagrS: 1.08e+04

39) 2.88
RealF: 125 LagrF: 175 RealS: 1.08e+04 LagrS: 1.02e+04
```

```
Консоль отладки Microsoft Visual Studio

37) 2.86
RealF: -89 LagrF: -41.6 RealS: 1.02e+04 LagrS: 1.08e+04

38) 2.87
RealF: 16.4 LagrF: 66.2 RealS: 1.08e+04 LagrS: 1.08e+04

39) 2.88
RealF: 125 LagrF: 175 RealS: 1.08e+04 LagrS: 1.02e+04

40) 2.89
RealF: 230 LagrF: 276 RealS: 1.01e+04 LagrS: 8.78e+03

41) 2.9
RealF: 325 LagrF: 364 RealS: 8.73e+03 LagrS: 6.72e+03

42) 2.91
RealF: 403 LagrF: 431 RealS: 6.68e+03 LagrS: 4.09e+03

43) 2.92
RealF: 457 LagrF: 472 RealS: 4.07e+03 LagrS: 1.04e+03

44) 2.93
RealF: 483 LagrF: 483 RealS: 1.03e+03 LagrS: -2.24e+03

45) 2.94
RealF: 477 LagrF: 460 RealS: -2.24e+03 LagrS: -5.52e+03

46) 2.95
RealF: 438 LagrF: 405 RealS: -5.54e+03 LagrS: -8.59e+03
```

```
46) 2.95
RealF: 438 LagrF: 405 RealS: -5.54e+03 LagrS: -8.59e+03

47) 2.96
RealF: 367 LagrF: 319 RealS: -8.62e+03 LagrS: -1.12e+04

48) 2.97
RealF: 267 LagrF: 207 RealS: -1.13e+04 LagrS: -1.32e+04

49) 2.98
RealF: 144 LagrF: 75.4 RealS: -1.32e+04 LagrS: -1.42e+04

50) 2.99
RealF: 5.29 LagrF: -66.9 RealS: -1.43e+04 LagrS: -1.43e+04

51) 3
RealF: -139 LagrF: -210 RealS: -1.44e+04 LagrS: -1.33e+04
```