

Интерполация по метода на Лагранж за реални функция и аргумент:

**Входни параметри:** масиви  $x$  и  $y$ ,  $x_1$  – точка, в която се търси решението

Изходни параметри:  $p$  – коефициенти на полинома и  $y_1$  стойността в  $x_1$

```
#include <iostream>
```

```
#include <vector>
```

```
using namespace std;
```

```
// Функция за изчисляване на стойността  $y_1$  в точка  $x_1$ 
```

```
double lagrange (int n, double x[], double y[], double x1) {
```

```
    double y1 = 0;
```

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

```
        double term = y[i];
```

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

```
            if (i != j) {
```

```
                term = term * (x1 - x[j]) / (x[i] - x[j]);
```

```
            }
```

```
        }
```

```
        y1 += term;
```

```
    }
```

```
    return y1;
```

```
}
```

```
// Функция за намиране на коефициентите на полинома
```

```
// Резултатът се връща в масив 'coeffs' с размер n
```

```
void Lcoeff (int n, double x[], double y[], vector<double>& coeffs) {
```

```
    fill(coeffs.begin(), coeffs.end(), 0.0);
```

```
// Временен масив за коефициентите на базовите полиноми
```

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

```
    vector<double> li(n, 0.0);
```

```
    li[0] = 1.0;
```

```
    double denominator = 1.0;
```

```
int current_degree = 0;
```

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

```
    if (i == j) continue;
```

```
        denominator *= (x[i] - x[j]);
```

```
// Умножение на полинома  $li$  с  $(x - x[j])$ 
```

```
for (int k = current_degree; k >= 0; k--) {
```

```
    li[k + 1] += li[k];
```

```
    li[k] *= -x[j];
```

```
}
```

```
current_degree++;
```

```

    }

    // Добавяне към крайните коефициенти
    for (int k = 0; k < n; k++) {
        coeffs[k] += (y[i] / denominator) * li[k];
    }
}
}

```

## // Примерна функция main()

```

int main() {
    int n = 3;
    double x[] = {0, 1, 2};
    double y[] = {1, 2, 4}; // Примерна функция  $y = 2^x$  (приблизително)
    double x1 = 1.5;

    // 1. Намиране на стойността y1
    double y1 = lagrange (n, x, y, x1);

    // 2. Намиране на коефициентите
    vector<double> coeffs(n);
    Lcoeff (n, x, y, coeffs);

    cout << "Stoinost v tochka x1 = " << x1 << " e y1 = " << y1 << endl;
    cout << "Koefficienti na polinoma (ot n-1 stepen kum 0):" << endl;
    for (int i = n - 1; i >= 0; i--) {
        cout << "a[" << i << "] = " << coeffs[i] << endl;
    }

    return 0;
}

```