

**Липецкий государственный технический университет**

**Факультет автоматизации и информатики**

**Кафедра автоматизированных систем управления**

**ЛАБОРАТОРНАЯ РАБОТА №1**

**По теме: Обработка массивов значений**

Студент

Первушин Е.А.

Группа

ПМ-23-2

Руководитель

Мирошников А.И.

Липецк 2023

**ЦЕЛЬ РАБОТЫ:** Освоить работу с массивами на языке C.

**ЗАДАНИЕ:** написать программу, выполняющую следующие действия:

1. Заполнение массива  $X$  вещественными числами, образующими арифметическую последовательность. Количество элементов в массиве ( $N$ ) и шаг последовательности ( $S$ ) заданы. Начальный элемент вводит пользователь.

2. Расчет заданной функции  $F(X)$  над элементами массива  $X$  и вывод значения функции пользователю. При выводе значение функции представить в экспоненциальном формате.

3. Расчет элементов массива  $Y$  по формуле  $Y_i = G(X_i)$  и вывод их пользователю в виде:

$y_1$ : 3.543

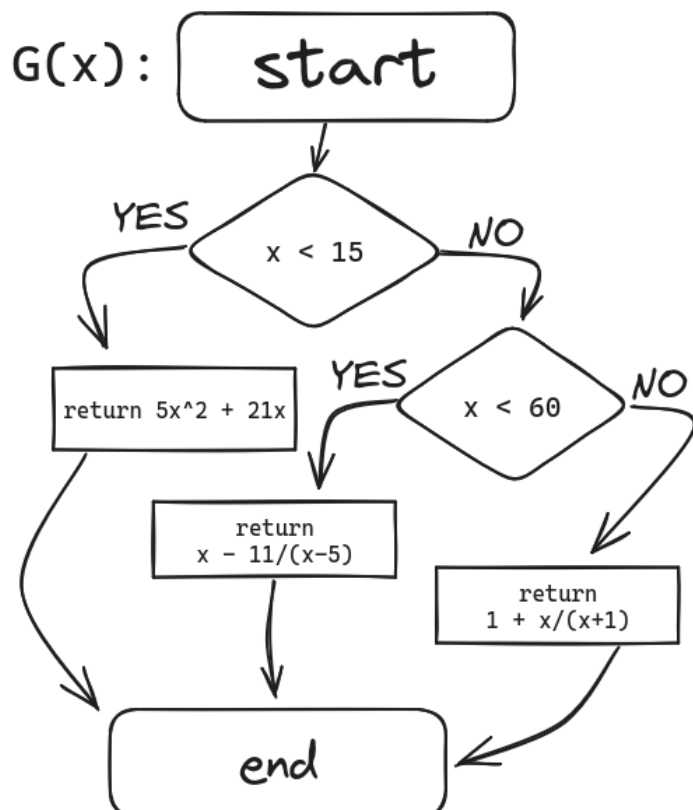
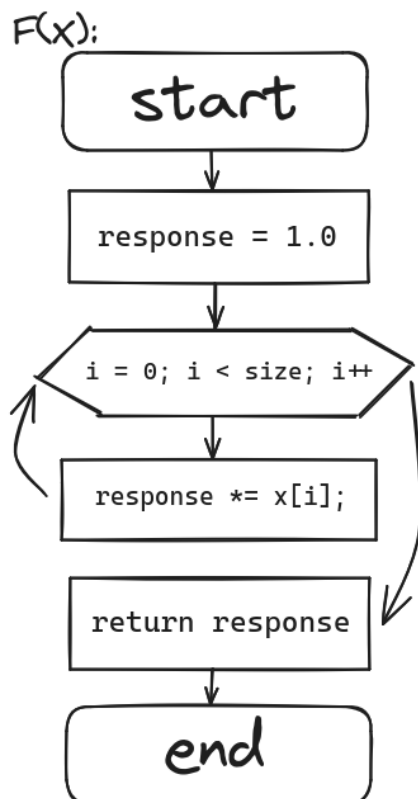
$y_2$ : 6.422

$y_3$ : -14.223

При выводе округлять значения  $Y_i$  до 3-х знаков после запятой.

## ВАРИАНТ 29

Блок-схемы алгоритмов:



## ТЕКСТ ПРОГРАММЫ

```
#include <stdio.h>
#include <string>
#include <cmath>

/* double to exponential */
std::string to_exponential(double d) {
    int exponent = (int) floor(log10(std::fabs(d)));
    double base = d * std::pow(10, -1.0 * exponent);

    char sign; if (exponent >= 0) { sign = '+'; } else { sign = '-'; }
    return std::to_string(base) + "E" + sign + std::to_string(exponent);
}

/* product of numbers */
double F(double *x, int size) {
    double response = 1.0;
    for (int i = 0; i < size; i++) { response *= x[i]; }
    return response;
}

double G(double x) {
    if (x < 15) { return 5*std::pow(x, 2) + 21*x; }
    else if (15 <= x && x < 60) { return x - (11/(x-5)); }
    else { return 1 + (x/(x+1)); }
}

int main() {
    const int N = 10;
    const double S = 39.5;

    /* task 1 */
    double X[N] = {};
    printf("Enter start number: "); scanf("%lf", &X[0]);
    for (int i = 1; i < N; i++) { X[i] = S * (double) i + X[0]; }

    /* task 2 */
    printf("\nF(x): %s\n\n", to_exponential(F(X, N)).c_str());

    /* task 3 */
    double Y[N] = {};
    for (int i = 0; i < N; i++) {
        Y[i] = G(X[i]);
        printf("y%d: %.3lf\n", i+1, Y[i]);
    }

    return 0;
}
```

## ПРИМЕР ВЫПОЛНЕНИЯ

```
→ labs git:(main) x g++ -o lab-1 lab-1.cpp
→ labs git:(main) x ./lab-1
Enter start number: -256.69
```

F(x): -1.827969E+19

```
y1: 324058.291
y2: 231296.490
y3: 154137.190
y4: 92580.390
y5: 46626.090
y6: 16274.291
y7: 1524.990
y8: 19.067
y9: 59.107
y10: 1.990
```

→ labs git:(main) ✕