

**Министерство науки и высшего образования
Российской Федерации**

**Федеральное государственное автономное
образовательное учреждение высшего образования**

**«Национальный исследовательский университет
ИТМО»**

**Факультет информационных технологий и
программирования**

Лабораторная работа № 10

Функции

Выполнил студент группы № М3109

Бабич Артём Антонович

Подпись:



Проверил:

Повышев Владислав Вячеславович

Санкт-Петербург
2020

Текст задания для 1 варианта

Задания:

1. *Реализовать функции нахождения наибольшего общего делителя и наименьшего общего кратного двух натуральных чисел.*
2. *На плоскости заданы своими координатами n точек. Разработать функцию, создающую массив размером $n \times (n - 1)$, элементами которого являются расстояния от каждой из точек до $n - 1$ других.*

Решение с комментариями

```
#include <stdio.h>
#include <math.h>
#include <locale.h>

// Задание 1

int GCD (int a, int b) {
    while (a != 0 && b != 0) {
        if (a > b) a = a % b;
        else b = b % a;
    }

    int result = a + b;
    return result;
}

int LCM (int a, int b) {
    int result = a / GCD(a, b) * b;
    return result;
}

void distances(int n, float arr[][n-1], float *crdx, float *crdy) {
    int i, j, c = 0;

    for (i = 0; i < n; i++) {
        for (j = 0; j < n ; j++) {
            if (i == 0 && j == 0) {
                arr[i][j] = sqrt(pow(crdx[i]-crdx[j + 1],2) + pow(crdy[i]-crdy[j + 1],2));
                j++;
                c++;
            }
            else{
                if (i == j){
                    c++;
                }
                else{
                    if (c == 0){
                        arr[i][j] = sqrt(pow(crdx[i]-crdx[j],2) + pow(crdy[i]-crdy[j],2));
                    }
                    else{
                        arr[i][j - 1] = sqrt(pow(crdx[i]-crdx[j],2) + pow(crdy[i]-crdy[j],2));
                    }
                }
            }
        }
        c = 0;
    }
}

int main() {
    setlocale (LC_ALL, "Ru");
    printf("Задание 1: \n \n");

    int a, b;

    printf("Введите первое число: ");
    scanf("%d", &a);
    printf("\n");
```

```

printf("Введите второе число: ");
scanf("%d", &b);
printf("\n");

int gcd = GCD(a, b);
printf("НОД введенных чисел: ");
printf("%d \n", gcd);

int lcm = LCM(a, b);
printf("НОК введенных чисел: ");
printf("%d \n \n", lcm);

// Задание 2

printf("Задание 2: ");
float x, y;
int n, i, j;
printf("\nВведите количество точек:");
scanf("%d", &n);
float crdx[n], crdy[n];
float d[n][n - 1];
for (i = 0; i < n; i++) {
    printf("Введите x%d and y%d: ", i, i);
    scanf("%f %f", &x, &y);
    crdx[i] = x;
    crdy[i] = y;
}
distances(n, d, crdx, crdy);
printf("\n");

printf("Ваш массив: \n");
for (i = 0; i < n; i++) {
    for (j = 0; j < n - 1; j++) {
        printf("%f ", d[i][j]);
    }
    printf("\n");
}

return 0;
}

```

Пошаговое решение:

1. Создаем целочисленную функцию для нахождения наибольшего общего делителя для двух чисел по методу Евклида;
2. Создаем целочисленную функцию для нахождения наименьшего общего кратного двух чисел;
3. Создаем функцию для нахождения расстояний между точками;
4. Запрашиваем числа для нахождения для них НОД и НОК;
5. Выводим результат вычислений на экран;
6. Запрашиваем с консоли количество точек на плоскости и их координаты;
7. Далее запрашиваем координаты для нужного количества точек;
8. Выводим массив с расстояниями;