

Правительство Российской Федерации

Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
«Высшая школа экономики»

Московский институт электроники и математики Национального
исследовательского университета «Высшая школа экономики»

Департамент прикладной математики

ОТЧЕТ

По лабораторной работе №1
«Работа 4. Обработка данных в одномерном массиве»

ФИО	Номер группы	Дата
Цыплаков Александр Александрович	БПМ214	06.12.2021

Москва-2021

Вариант 25

Задание:

Числовой массив B (тип массива указан в формулировке второго задания) содержит k элементов. Элементы массива и пороговые значения X, Y вводятся с клавиатуры. Написать подпрограммы создания массива и вывода его на экран. В первом задании требуется написать функцию нахождения соответствующего варианту максимального/минимального значения, а во втором – среднего арифметического указанных в условии элементов («между» понимать строго – не включая найденные позиции).

Оба задания реализовать в одной программе.

25	$\min(b_1, \dots, b_k)$ для $ b_i > X$	Среднее арифметическое элементов, расположенных до последнего максимального элемента. Массив вещественный.
----	---	--

1)Решение:

```
#include <stdio.h> //библиотека, содержащая функции форматированного ввода и вывода переменных
#include <locale.h> //библиотека, отвечающая за локализацию (setlocale)
#include <math.h> //библиотека, содержащая математические выражения(fabs)
#include <stdlib.h> //библиотека, отвечающая за выделение памяти
#define SIZE 10 //константное значение

void input(double* B, int K) //функция заполнения массива, в скобках указаны аргументы, передаваемые в функцию
{
    //printf("Введите элементы массива:\n");
    for (int i = 0; i < K; i++) //цикл, для заполнения массива, с индексами от i до K
    {
        printf("b[%d]=", i); //форматированный вывод индекса элемента массива
        scanf("%lf", &B[i]); //форматированный ввод элементов массива
    }
}

double f1(double* B, int K, double X) //функция нахождения минимума, по модулю большего порогового значения, в
скобках указаны аргументы, передаваемые в функцию
{
    double min = 0, k = 0, l = 0; //инициализация необходимых переменных
    for (int i = 0; i < K; i++) //перебор элементов массива, с помощью цикла
    {
        if (fabs(B[i]) > X) //условный оператор, если элемент массива по модулю больше порогового, то заходим внутрь
        оператора
        {
            if ((min == 0) && (l == 0)) //присвоение переменной значение первого элемента массива, который прошел
            отбор
            {
                min = B[i]; //присвоение переменной значение элемента массива
                l++; //необходимо, чтобы в дальнейшем элементы не удовлетворяли условию оператора
            }
            if (B[i] <= min) //нахождение минимума
            {
                min = B[i]; //присвоение минимуму значение элемента массива
            }
        }
    }
}
```

```

    }
}

if((min == 0) && (l == 0)) //если минимума нет, то присваиваем ноль
{
    min = 0;
}

return min; //возвращение нужного значения
}

double f2(double* B, int K) //функция нахождения среднего арифметического среди элементов до последнего
максимума, в скобках указаны аргументы, передаваемые в функцию
{
    double max, srznach = 0, sum = 0; //объявление/инициализация переменных, необходимых для решения
    int index_max = 0; //инициализация переменной
    max = B[0]; //присвоение переменной значение самого первого элемента массива
    for (int i = 0; i < K; i++) //перебор элементов массива
    {
        if(B[i] >= max) //нахождение максимума
        {
            max = B[i]; //присваиваем переменной значения подходящего элемента массива
            index_max = i; //запоминаем индекс этого элемента
        }
    }

    if (index_max == 0) //если индекс максимального элемента равен 0, то
    {
        srznach = 0; //присваиваем 0
    }
    else //в других случаях
    {
        for(int i = 0; i < index_max; i++) //перебор элементов массива до последнего максимального
        {
            sum += B[i]; //вычисление суммы
        }
        srznach = sum/index_max; //подсчет среднего арифметического
    }
    return srznach; //возвращаемое значение
}

int main() //главная функция
{
    double B[SIZE]; //объявление массива, под который выделяем память для SIZE элементов, тип массива - double
    double X, Y; //объявлении переменных, пороговые значения
    int K; //объявлении переменной для ограничения количества элементов массива
    double MIN, SRZNACH; //объявление переменных

    printf("Введите количество элементов массива:"); //форматированный вывод строки
    scanf("%d", &K); //форматированный ввод количества элементов массива

    setlocale(LC_ALL, "rus"); //подключение русского языка

```

```

setlocale(LC_NUMERIC, "eng"); //замена запятой на точку

printf("Введите элементы массива:\n"); //форматированный вывод строки
input(B, K); //вызов функции заполнения массива

printf("Введите пороговые значения:\n"); //форматированный вывод строки
printf("X = "); //форматированный вывод строки
scanf("%lf", &X); //форматированный ввод порогового значения
printf("Y = "); //форматированный вывод строки
scanf("%lf", &Y); //форматированный ввод переменной

MIN = f1(B, K, X); //присваиваем переменной результат функции (результат, который является возвращающим
значением функции)
printf("\t(1)Минимальный элемент из тех, что больше X: %lf\n", MIN); //форматированный вывод строки и
переменной

SRZNACH = f2(B, K); //присваиваем переменной результат функции (результат, который является возвращающим
значением функции)
printf("\t(2)Среднее арифметическое значение элементов до последнего максимального: %lf\n", SRZNACH);
return 0; //форматированный вывод строки и переменной
}

```

В самом начале подключаем библиотеки, необходимые для решения задания. Далее в главной функции объявляем массив, для которого выделяется память под SIZE элементов (константное значение, которое инициализируем после подключения всех библиотек). Заполнение массива происходит в функции input, которая не возвращает значение.

1)Первая часть задания заключается в нахождении минимального элемента массива, который при этом по модулю больше вводимого порогового значения. Все необходимые операции проводятся в функции f1.

Сначала инициализируем все необходимые переменные. Далее начинаем перебор элементов массива с помощью цикла for, в теле цикла прописываем нужное нам условие, минимум по модулю больше порогового значения, с помощью условного оператора: `if(fabs(B[i]) > x)`. В теле условного оператора первым делом присваиваем переменной min значение элемента массива, который удовлетворяет условию, написанному ранее, чтобы в дальнейшем не заходить в этот оператор и не присваивать значение, что повлекло бы неправильный ответ, увеличиваем переменную i на 1, тогда условие этого оператора больше не будет выполняться. Далее находим минимальный элемент массива, с помощью еще одного условного оператора.

Нужно сделать поправку, что если у нас нет элементов, которые больше по модулю порогового значения. Если у нас нет таких элементов, значит у нас min так и остается 0, а переменная i, не изменится, так как мы не меняли их,

следовательно прописываем эти условия в условном операторе и в таком случае присваиваем переменной `min` значение 0. У функции возвращаемое значение – переменная `min`.

В главной функции объявляем переменную `MIN`, которой присваиваем результат функции `f1`, и выводим в консольное окно.

2) Вторая часть задания заключается в том, чтобы найти среднее арифметическое элементов, расположенных до последнего максимального элемента массива.

Данное задание можно разбить на два пункта а и б:

а) Сначала нужно найти максимум, среди элементов массива. Для этого переменной `max` присваиваем значение первого элемента массива и затем перебираем элементы массива с помощью цикла `for`. Внутри цикла создаем условный оператор, благодаря которому и будем искать максимум, при этом запоминаем индекс максимального элемента.

Если индекс максимального элемента равен 0, то есть максимальный элемент расположен первым, то сразу присваиваем переменной `srznach` значение 0, так как нам нужно посчитать среднее элементов ДО, а максимум расположен первым. В других случаях находим среднее арифметическое.

б) Нахождение среднего арифметического среди элементов до последнего максимального. Перебор элементов массива, начиная с нулевого, заканчивая индексом максимального элемента, и складываем все эти переменные в переменную `sum`. Далее в переменную `srznach` присваиваем результат выражения `sum/index_max`, что и будет нашим средним арифметическим. Возвращаемым значением данной функции является переменная `srznach`.

В главной функции объявляем переменную `SRZNACH`. Ей будем присваивать возвращаемое значение функции `f2` и выводить его в консольное окно