

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ
КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М. А. БОНЧ-БРУЕВИЧА»
(СПбГУТ)

ОТЧЁТ
ЛАБОРАТОРНАЯ РАБОТА №12

Указатели на функции

Руководитель,
старший преподаватель

подпись, дата

Помогалова А. В.

Исполнитель,
группа ИКПИ-33

подпись, дата

Коньков М. Д.

Санкт-Петербург 2024

УСЛОВИЯ РАБОТЫ

Вариант № 12:

Необходимо выполнить заданную обработку файла:

12	Написать функцию пользователя, позволяющую найти среднее значение произвольной функции одного переменного, аргумент которой задан массивом из «n» элементов вещественного типа.
----	---

Общая формулировка:

Необходимо написать программу с использованием указателя на функцию. Эта программа будет получать на вход числа от пользователя и затем возводить в квадрат каждое, после чего найдёт среднее арифметическое этих квадратов, выведя результат на экран.

Переменные:

N	Обозначение в задаче	Идентификатор	Назначение
1	<i>again</i>	<i>Char again</i>	Итератор программы
2	<i>n</i>	<i>Int n</i>	Количество чисел
3	<i>avg</i>	<i>Double avg</i>	Число строк
4	<i>sum</i>	<i>Double sum</i>	Сумма
5	<i>arr</i>	<i>Double arr[]</i>	Массив чисел
6	<i>x</i>	<i>Double x</i>	Параметр квадрата
7	<i>i</i>	<i>int i</i>	Итератор цикла

1 Общий алгоритм решения

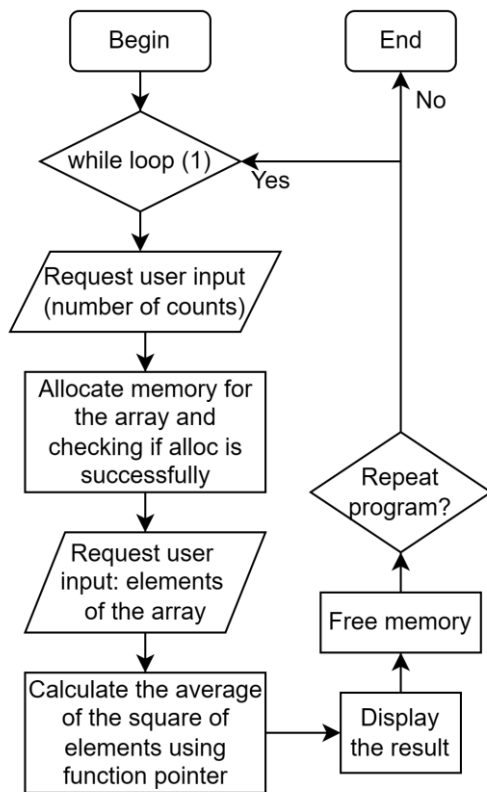
Процесс выполнения лабораторной работы можно разбить на **подзадачи**:

Хорошо, вот пошаговый отчет для этой программы:

1. Включение необходимых заголовочных файлов:
2. Определение типа функции `FuncPointer`, который представляет собой указатель на функцию, принимающую один аргумент типа `double` и возвращающую `double`.
3. Определение функции `average`, которая принимает указатель на функцию `func`, массив `arr` и его длину `n`. Функция вычисляет среднее значение функции `func` на элементах массива `arr`.
4. Определение примерной функции `square`, которая принимает `double x` и возвращает квадрат этого числа.
5. Основная функция `main`:
 - В бесконечном цикле `while(1)` программа выполняет следующие шаги:
 - Запрашивает у пользователя количество элементов `n` в массиве.
 - Выделяет память под массив с помощью `malloc`.
 - Запрашивает у пользователя каждый элемент массива и сохраняет его в выделенной памяти.
 - Вызывает функцию `average` с функцией `square` и массивом, затем выводит среднее значение квадратов элементов массива.
 - Освобождает выделенную память под массив с помощью `free`.
 - Спрашивает пользователя, желает ли он повторить программу. Если ответ не 'Y' или 'y', программа завершается с помощью `break`.

Этот код позволяет пользователю вводить произвольное количество элементов в массиве и вычисляет среднее значение квадратов этих элементов. После каждого вычисления программа спрашивает пользователя, хочет ли он повторить процесс или завершить выполнение программы.

1.1 Общий алгоритм решения



1.2 Тестирование:

Для тестирования выберем 5 чисел: 1, 2, 3, 4 и 5. Итог будет следующим:

11. Так можно сделать с любым количеством произвольных действительных чисел, которые пользователь будет вводить вручную.

ПРИЛОЖЕНИЕ А

ЛИСТИНГ ПРОГРАММЫ LAB12.C (C)

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

// Объявление типа функции
typedef double (*FuncPointer)(double);

// Функция для нахождения среднего значения функции на массиве
double average(FuncPointer func, double arr[], int n) {
    double sum = 0.0;
    for (int i = 0; i < n; i++) {
        sum += func(arr[i]); // Вызываем функцию, переданную через указатель
    }
    return sum / n;
}

// Пример функции, которую можно использовать
double square(double x) {
    return x * x;
}

int main() {

    while(1) {
        char again;
        int n;

        // Запрос у пользователя количества чисел в массиве
        printf("Enter the number of elements in the array: ");
        scanf("%d", &n);

        // Выделение памяти под массив
        double *arr = (double*)malloc(n * sizeof(double));

        // Проверка на успешное выделение памяти
        if (arr == NULL) {
            printf("Memory allocation failed!\n");
            return 1;
        }
    }
}
```

```

    }

    // Запрос у пользователя каждого числа и добавление его в массив
    printf("Enter the elements of the array:\n");
    for (int i = 0; i < n; i++) {
        scanf("%lf", &arr[i]);
    }

    // Вызываем функцию average и передаем в нее указатель на функцию square
    double avg = average(square, arr, n);
    printf("Average of the square of the array elements: %lf\n", avg);

    // Освобождение памяти, выделенной под массив
    free(arr);

    // Проверка на повторение
    printf("\n\033[0;32mRepeat program? \033[0m(Y, y -- yes / other -- no)
\nAnswer: ");
    scanf(" %c", &again);
    if ( (again != 'Y') && (again != 'y') )
    {
        break;
    }
}
return 0;

```

ПРИЛОЖЕНИЕ Б

РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ LAB12.C (C)

Вывод командной строки:

```
D:\University\Programming\NotGit\LabsC1.2\Lab12>Lab12.exe
Enter the number of elements in the array: 5
Enter the elements of the array:
1
2
3
4
5
Average of the square of the array elements: 11.000000

Repeat program? (Y, y -- yes / other -- no)
Answer: no

D:\University\Programming\NotGit\LabsC1.2\Lab12>
```