МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ассистент |  |  |  | М.А. Мурашова |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 1 |
| «Анализ сложности алгоритмов » |
| по курсу: Структуры и алгоритмы обработки данных |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4134К |  |  |  | Опарин С.Н. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

1. **Цель работы:**

Целью работы является изучение методов и получение практических навыков анализа сложности алгоритмов.

1. **Задание**

Используя память, пропорциональную n, хранить массив целых чисел A, содержащий n элементов.

Элементы массива A могут принимать случайные значения от -((n div 2) - 1) до (n div 2). То есть, если в массиве хранится 10 элементов, то эти элементы должны быть в диапазоне от -4 до 5.

Разработать алгоритм, который осуществляет заполнение массива A случайными значениями, и по выбору пользователя выполняет одну из двух функций. Состав выполняемых функций и требования к теоретической временной сложности этих функций определяется вариантом задания. Варианты задания приведены в таблице 1.

1. **Вариант задания**

****

1. **Листинг программы, реализующей алгоритм:**

#include <iostream>

**using** **namespace** std;

**int** \*fill\_mas\_with\_pos\_nums(**int** \*arr, **int** N){

**for**(**int** i = 0; i < N; i++){

**if** (arr[i] < 0){

arr[i] = abs(arr[i]);

}

**else**{

**continue**;

}

cout << arr[i] << endl;

}

**return** arr;

}

**void** output\_even\_nums(**int** count){

cout << "Количество четных чисел в массиве: " << count << endl;

}

**int** main(){

cout << "Введите длину массива: ";

**int** N, count\_of\_even\_nums = 0, choose;

cin >> N;

**int** \*arr = **new** **int**[N];

cout << "Введите массив поэлементно: " << endl;

**for**(**int** i = 0; i < N; i++){

cin >> arr[i];

**while**((**int**((-N / 2) - 1) >= arr[i]) **or** (arr[i] >= **int**(N / 2))){

cout << "Введено значение не из диапазона " << "от " << **int**((-N / 2) - 1) << " до " << **int**(N / 2) << endl;

cout << "Введите новое: ";

cin >> arr[i];

}

}

cout << "Выберите функцию, которую хотите запустить" << endl;

cout << "Введите 1, если первую и введите 2, если вторую: ";

cin >> choose;

**if** (choose == 1){

arr = fill\_mas\_with\_pos\_nums(arr, N);

}

**else**{

**for**(**int** i = 0; i < N; i++){

cout << arr[i] << endl;;

**if**(arr[i] % 2 == 0){

count\_of\_even\_nums += 1;

}

}

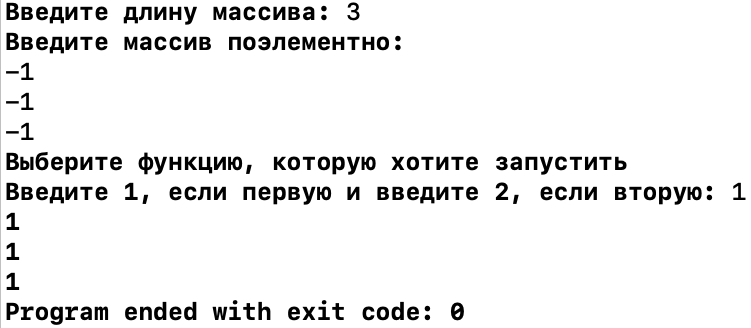
output\_even\_nums(count\_of\_even\_nums);

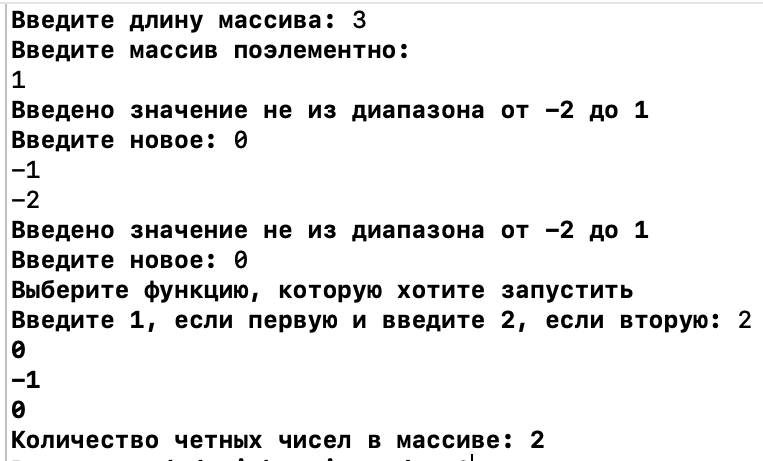
}

**return** 0;

}

Примеры работы:





1. **Расчет пространственной и временной сложности алгоритма:**

***Теперь подсчитываем теоретические сложности алгоритма.*** Разработанный алгоритм использует следующие данные:

− один массив размерностью n;

− шесть переменные целого типа.

Значит, пространственная сложность алгоритма определяется следующим образом:

где – константа, характеризующая объем памяти, отводимый под переменную целого типа.

***Временная сложность рассчитывается таким образом:***

1. **Расчет теоретической пространственной и теоретической временной сложности алгоритма:**

***Теоретическая пространственная сложность алгоритма составляет:***

***Теоретическая временная сложность алгоритма составляет:***

1. **Вывод**

В ходе лабораторной работы я изучил методы анализа сложности алгоритмов, а также применил эти знания на практике.

Лабораторная выполнена качественно и хорошо.