

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

ассистент

должность, уч. степень, звание

подпись, дата

Синёв Н. И.

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №5

ОДНОМЕРНЫЕ МАССИВЫ

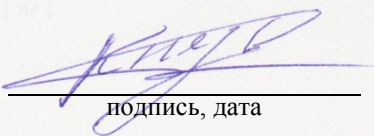
КУРС 1

по курсу: ОСНОВЫ ПРОГРАММИРОВАНИЯ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

1941



подпись, дата

Князюк Р.А.

инициалы, фамилия

Санкт-Петербург

2020

1. Постановка задачи.

В массиве из 10 целых чисел определить номер пары соседних элементов с максимальной суммой.

2. Формализация.

- **Список переменных в данной программе:**

- Символьные константы:
SIZE – 10 (кол-во элементов в массиве);
- Массив типа **int**:
mass[SIZE] – массив для хранения целых чисел(кол-во эл. *SIZE*);
- Переменные типа **int**:
max_sum – переменная, которая хранит максимальную сумму пары;
num_pair – переменная для хранения номера элемента массива, который при суммировании со следующим дает максимальную сумму;
i – счетчик ,обозначающий номер элемента массива (используется для циклов);

- Все данные вводятся с клавиатуры.
- Ввод массива производится циклически, пока счетчик (*i*) не станет равен 10. (Так как количество итераций известно, будет использован цикл **for**)
- Будем считать, что в массиве существует хотя бы одна пара элементов, дающая максимальную сумму.
- Для хранения номера используем одну переменную, так как по условию необходимо найти пару *соседних* элементов. Следовательно, порядковый номер второго элемента пары можно найти, прибавив 1 к номеру первого элемента пары.
- Будем считать, что порядковый номер числа в массиве соответствует номеру элемента массива увеличенному на 1.(то есть 0 и 1 элементы массива это 1 и 2 порядковый номер числа)
- После ввода массива присвоим максимальному значению сумму первой пары элементов.
- Поиск максимальной пары производим циклически, начиная с 1 элемента массива (так как *max_sum* хранит сумму первой пары элементов массива), пока *i* < *SIZE-1*. В цикле сравниваем *max_sum* с суммой текущей пары. Если *max_sum* *меньше* , то *max_sum* присвоить значение текущей пары и номеру пары присвоить значение счетчика *i*.

- В конце программа должна вывести максимальную сумму пары и порядковые номера элементов, которые дают эту сумму.
- Операционная система: Windows 7 Максимальная
- Написание кода: Visual Studio Code
- Компилятор: MinGW GCC-8.2.0-5
- Запуск программы через *.exe

Таблица 1 – Тестовые примеры

Пример	№1	№2
Данные (точка с запятой отделяет введенные данные)	1;2;3;4;5;6;7;8;9;10	1;124;14;442;1; 1;2;4;5;10
Результат	9 и 10 номер элементов; Максимальная сумма 19	3 и 4 номер элементов; Максимальная сумма 456

3. Алгоритмизация. Схема алгоритма.

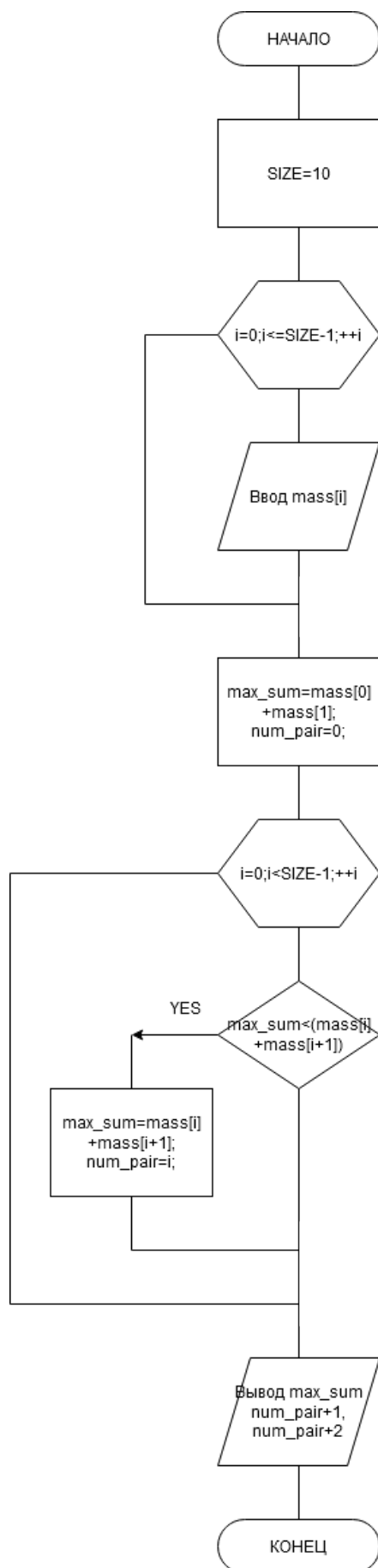


Рисунок 1 – Блок-схема алгоритма

4. Код программы на Си.

```
#include <stdio.h> //для scanf
#include <stdlib.h> //для system("pause")

#define SIZE 10

int main(void){
    int mass[SIZE];
    int i,max_sum,num_pair;
    //Ввод массива
    for(i=0;i<=SIZE-1;++i){
        printf("num = %2d",i+1);
        printf(" mass[%d] = ",i);
        scanf("%d",&mass[i]);
    }

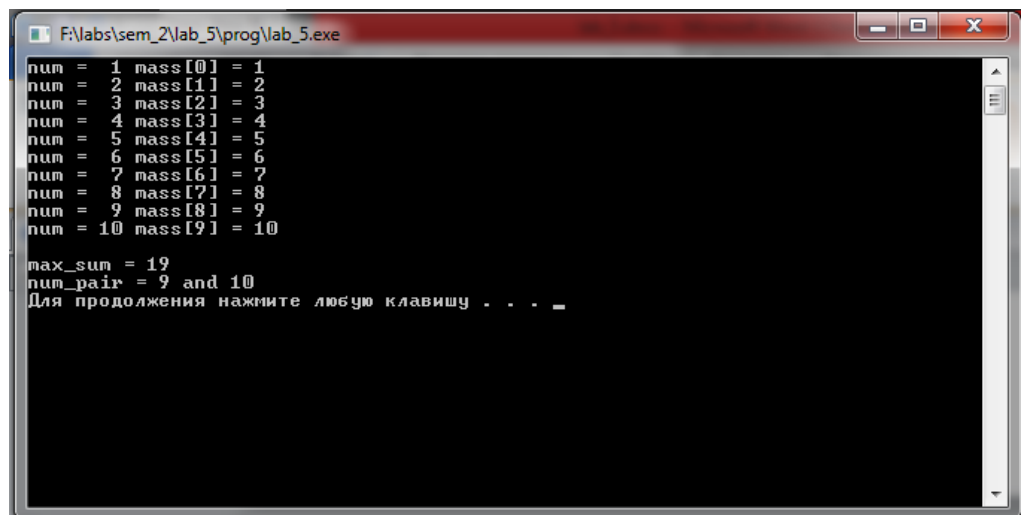
    max_sum=mass[0]+mass[1];
    num_pair=0;

    //поиск максимальной суммы и номера пары
    for(i=1;i<SIZE-1;++i){
        if(max_sum<(mass[i]+mass[i+1])){
            max_sum=mass[i]+mass[i+1];
            num_pair=i;
        }
    }

    //вывод номера пары и максимальной суммы
    printf("\nmax_sum = %d\nnum_pair = %d and %d\n",max_sum,num_pair+1,num
_pair+2);

    system("pause");
    return 0;
}
```

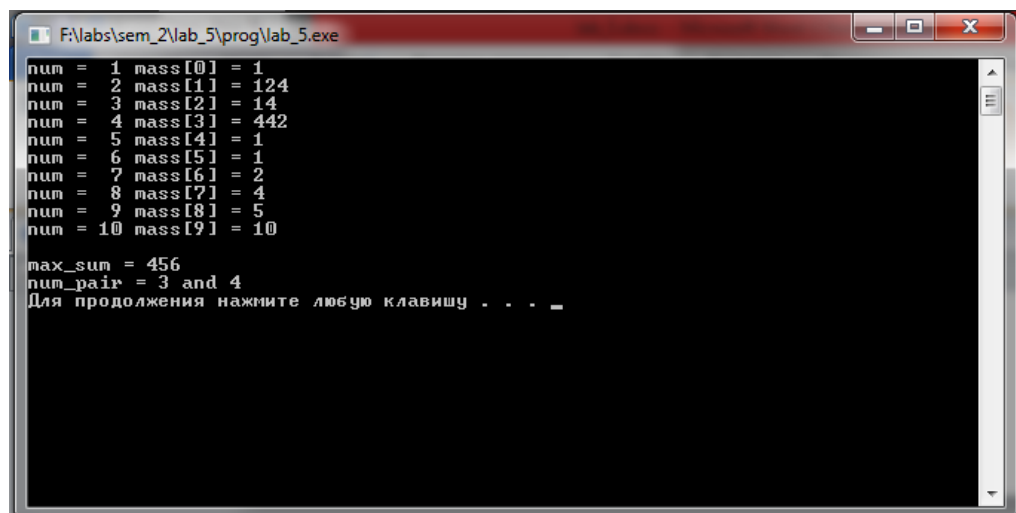
5. Проверка работы программы на тестовых примерах.



```
F:\labs\sem_2\lab_5\prog\lab_5.exe
num = 1 mass[0] = 1
num = 2 mass[1] = 2
num = 3 mass[2] = 3
num = 4 mass[3] = 4
num = 5 mass[4] = 5
num = 6 mass[5] = 6
num = 7 mass[6] = 7
num = 8 mass[7] = 8
num = 9 mass[8] = 9
num = 10 mass[9] = 10

max_sum = 19
num_pair = 9 and 10
Для продолжения нажмите любую клавишу . . . _
```

Рисунок 2 – Тестовый пример 1



```
F:\labs\sem_2\lab_5\prog\lab_5.exe
num = 1 mass[0] = 1
num = 2 mass[1] = 124
num = 3 mass[2] = 14
num = 4 mass[3] = 442
num = 5 mass[4] = 1
num = 6 mass[5] = 1
num = 7 mass[6] = 2
num = 8 mass[7] = 4
num = 9 mass[8] = 5
num = 10 mass[9] = 10

max_sum = 456
num_pair = 3 and 4
Для продолжения нажмите любую клавишу . . . _
```

Рисунок 3 – Тестовый пример 2

6. Вывод.

Тестовые примеры (5 пункт отчета) доказывают правильность работы алгоритма и программы. Таким образом, программа способна определить номер пары соседних элементов с максимальной суммой в массиве из 10 элементов, а также вывести максимальную сумму и номера элементов на экран.