|  |
| --- |
| 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 |  |  |  | Синёв Н. И. |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №5 |
| ОДНОМЕРНЫЕ МАССИВЫКУРС 1 |
| по курсу: ОСНОВЫ ПРОГРАММИРОВАНИЯ |
|  |
|  |

****РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 1941 |  |  |  | Князюк Р.А. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург

2020

1. **Постановка задачи.**

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

1. **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;

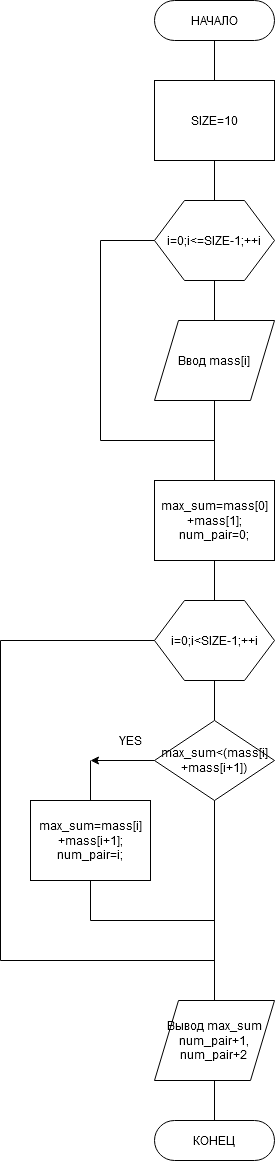
***i*** – счетчик ,обозначающий номер элемента массива (используется для циклов);

* 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 паре 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 |

1. **Алгоритмизация. Схема алгоритма.**

****

**Рисунок 1 – Блок-схема алгоритма**

1. **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

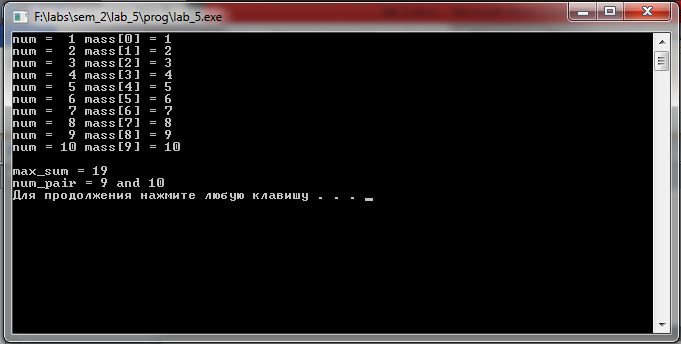
    printf("\nmax\_sum = %d\nnum\_pair = %d and %d\n",max\_sum,num\_pair+1,num\_pair+2);

    system("pause");

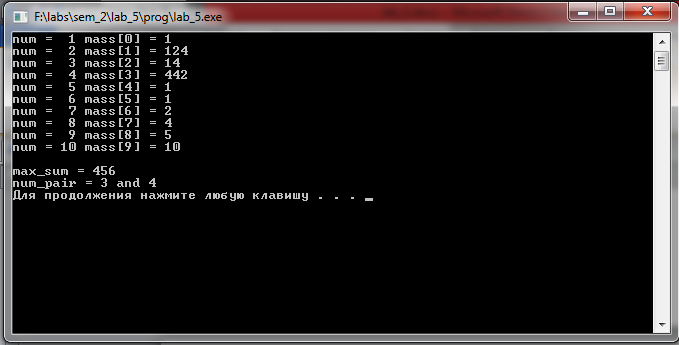
    return 0;

}

1. **Проверка работы программы на тестовых примерах.**

****

**Рисунок 2 – Тестовый пример 1**

****

**Рисунок 3 – Тестовый пример 2**

1. **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.