



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

« МИРЭА Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий

Кафедра Вычислительной техники

УЧЕБНОЕ ЗАДАНИЕ

по дисциплине

« Объектно-ориентированное программирование»

Наименование задачи:

« Задание 2_1_3 »

С тудент группы

ИКБО-27-21

Родионов А.А.

Руководитель практики

Ассистент

Морозов В.А.

Работа представлена

«__»_____ 2022 г.

(подпись студента)

Оценка

(подпись руководителя)

Москва 2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
Постановка задачи.....	5
Метод решения.....	7
Описание алгоритма.....	9
Блок-схема алгоритма.....	14
Код программы.....	18
Тестирование.....	21
ЗАКЛЮЧЕНИЕ.....	22
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ (ИСТОЧНИКОВ).....	23

ВВЕДЕНИЕ

Постановка задачи

Создать объект, который обрабатывает массив целых чисел не более 10 элементов.

Количество элементов определяются в момент конструирования объекта.

Объект обладает следующей функциональностью:

- в конструкторе считывает значение количества элементов массива, выводит значение количества элементов;
- считывает значения элементов массива;
- выводит значения элементов массива;
- разворачивает последовательность значений элементов массива.

Написать программу, которая:

1. Создает объект и в конструкторе считывает количество элементов массива;
2. Считывает элементы массива;
3. Выводит значения элементов массива согласно исходной последовательности;
4. Разворачивает элементы массива;
5. Выводит значения элементов массива согласно новому их порядку следования.

Описание входных данных

Первая

строка:

целое число в десятичном формате.

Вторая

строка:

последовательность целых чисел в десятичном формате разделенных пробелом.

Описание выходных данных

Первая

строка:

N = «количество элементов»

Вторая строка (исходный порядок следования элементов):

Значения элементов массива, значение каждого элемента занимает 5 позиции,

выравнивание по правому краю.

Третья строка (порядок следования элементов после разворота):

Значения элементов массива, значение каждого элемента занимает 5 позиции,

выравнивание по правому краю.

Метод решения

Для выполнения задачи потребовалось :

- Переменная типа integer
- Объект ввода/вывода потока данных cin/cout (iostream)
- Условный оператор if
- Цикл for
- Класс Vector :

Поля:

Модификатор доступа : private :

Наименование : s

функционал : хранит длину массива

Наименование : mas

функционал : указатель на массив mas

Модификатор доступа : public :

Наименование : Vector

функционал : конструктор класса

Наименование : ~Vector

функционал : деструктор класса

Наименование : reverse

функционал : переворачивает массив

Наименование : display

функционал : выводит массив

Наименование : set_element

Параметры : int e, int i

функционал : устанавливает значение элемента массива с индексом i
значение e

Наименование : get_length

Возвращаемый тип : int

функционал : возвращает размер массива

Описание алгоритма

Согласно этапам разработки, после определения необходимого инструментария в разделе «Метод», составляются подробные описания алгоритмов для методов классов и функций.

Класс объекта: Vector

Модификатор доступа: public

Метод: Vector

Функционал: Конструктор класса

Параметры: int len

Возвращаемое значение: void

Алгоритм метода представлен в таблице 1.

Таблица 1. Алгоритм метода Vector класса Vector

№	Предикат	Действия	№ перехода	Комментарий
1		Инициализация переменной типа integer int len	2	
2		Ввод значения len	3	
3	!(len >= 1 && len <= 10)		4	
			5	
4		Вывод "Invalid input"	5	
5		this->s = len	6	
6		this->mas = new int[len]	Ø	

Класс объекта: Vector

Модификатор доступа: public

Метод: ~Vector

Функционал: Деструктор

Параметры: нет

Возвращаемое значение: void

Алгоритм метода представлен в таблице 2.

Таблица 2. Алгоритм метода ~Vector класса Vector

№	Предикат	Действия	№ перехода	Комментарий
1		Удаление mas	Ø	

Класс объекта: Vector

Модификатор доступа: public

Метод: reverse

Функционал: Переворачивает массив

Параметры: нет

Возвращаемое значение: void

Алгоритм метода представлен в таблице 3.

Таблица 3. Алгоритм метода reverse класса Vector

№	Предикат	Действия	№ перехода	Комментарий
1		for(int i = 0; i < s / 2; i++)	2	
2		Инициализация целочисленной переменной buf	3	

3		buf = mas[s - i - 1];	4	
4		mas[s - i - 1] = mas[i];	5	
5		mas[i] = buf;	Ø	

Класс объекта: Vector

Модификатор доступа: public

Метод: display

Функционал: Выводит массив

Параметры: нет

Возвращаемое значение: void

Алгоритм метода представлен в таблице 4.

Таблица 4. Алгоритм метода display класса Vector

№	Предикат	Действия	№ перехода	Комментарий
1		for(int i = 0; i < this->s;i++)	2	
2		int n,l	3	
3		n = mas[i]	4	
4		l = 0	5	
5	n < 0		6	
			8	
6		l++	7	
7		n *= -1	8	
8		while(n > 0)	9	
9		n /= 10	10	
10		l++	11	
11	5 - l < 0		12	
			13	
12		Вывод " "	15	
13		for(int s = 0; s < 5 - l;s++)	14	
14		Вывод " "	15	

15		Вывод mas[i]	Ø	
----	--	--------------	---	--

Класс объекта: Vector

Модификатор доступа: public

Метод: set_element

Функционал: Устанавливают значение массива по индексу

Параметры: int e, int i

Возвращаемое значение: void

Алгоритм метода представлен в таблице 5.

Таблица 5. Алгоритм метода set_element класса Vector

№	Предикат	Действия	№ перехода	Комментарий
1		mas[i] = e	Ø	

Функция: main

Функционал: Главная функция программы

Параметры: нет

Возвращаемое значение: void

Алгоритм функции представлен в таблице 6.

Таблица 6. Алгоритм функции main

№	Предикат	Действия	№ перехода	Комментарий
1		Создание объекта Vector	2	

		Vector vec;		
2	(vec.get_length() >= 1 && vec.get_length() <= 10)		3	
			Ø	
3		for(int i = 0;i < s;i++)	4	
4		Инициализация целочисленной переменной e	5	
5		Ввод значения e	6	
6		vec.set_element(e, i)	7	
7		Выход из цикла	8	
8		Вывод "N = " s endl	9	
9		vec.display();	10	
10		Вывод "\n"	11	
11		vec.reverse()	12	
12		vec.display()	Ø	

Класс объекта: Vector

Модификатор доступа: public

Метод: get_length

Функционал: возвращает длину массива свойства класса

Параметры: нет

Возвращаемое значение: integer

Алгоритм метода представлен в таблице 7.

Таблица 7. Алгоритм метода get_length класса Vector

№	Предикат	Действия	№ перехода	Комментарий
1		return this->s	Ø	

Блок-схема алгоритма

Представим описание алгоритмов в графическом виде на рисунках ниже.

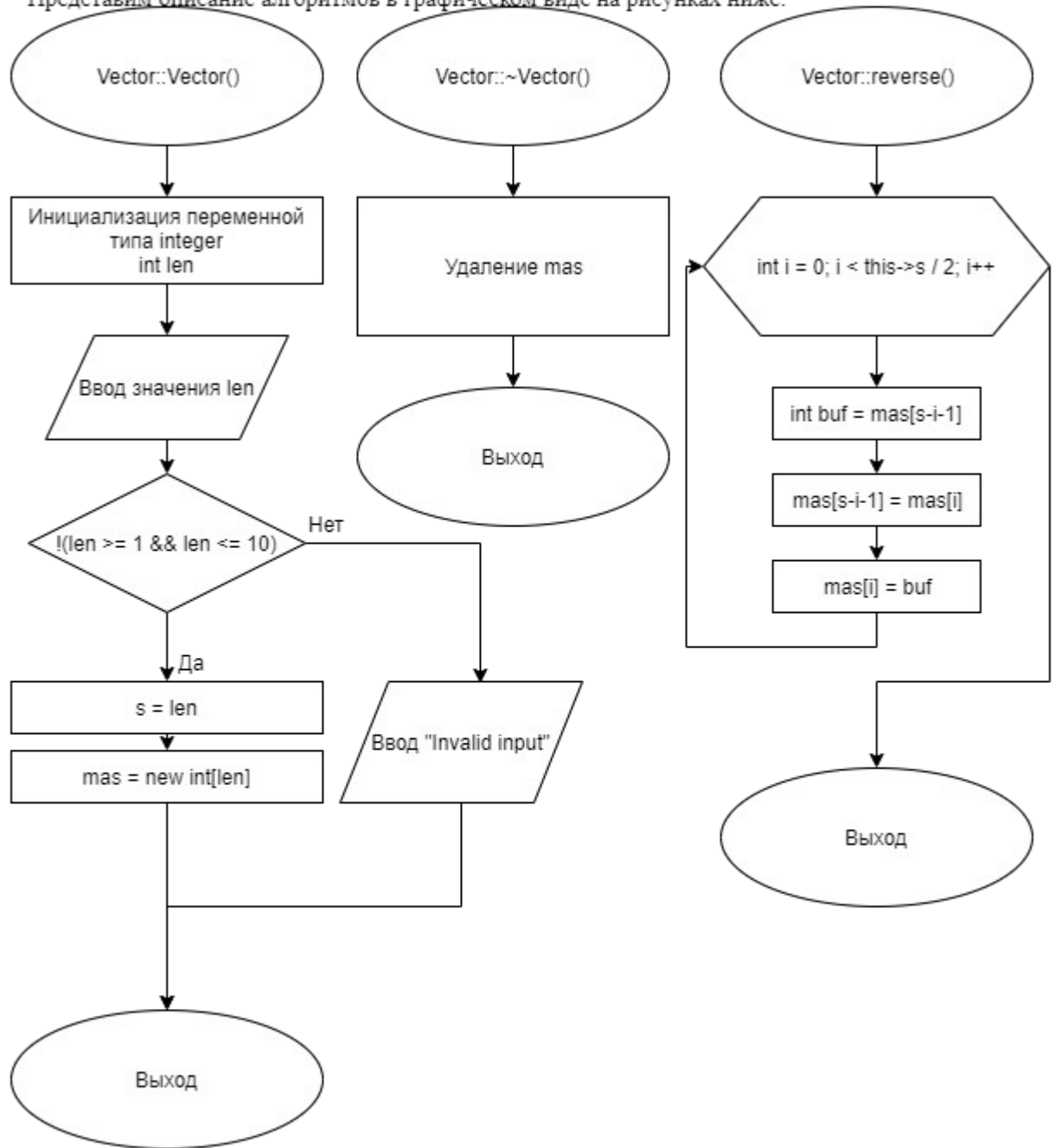


Рис. 1. Блок-схема алгоритма.

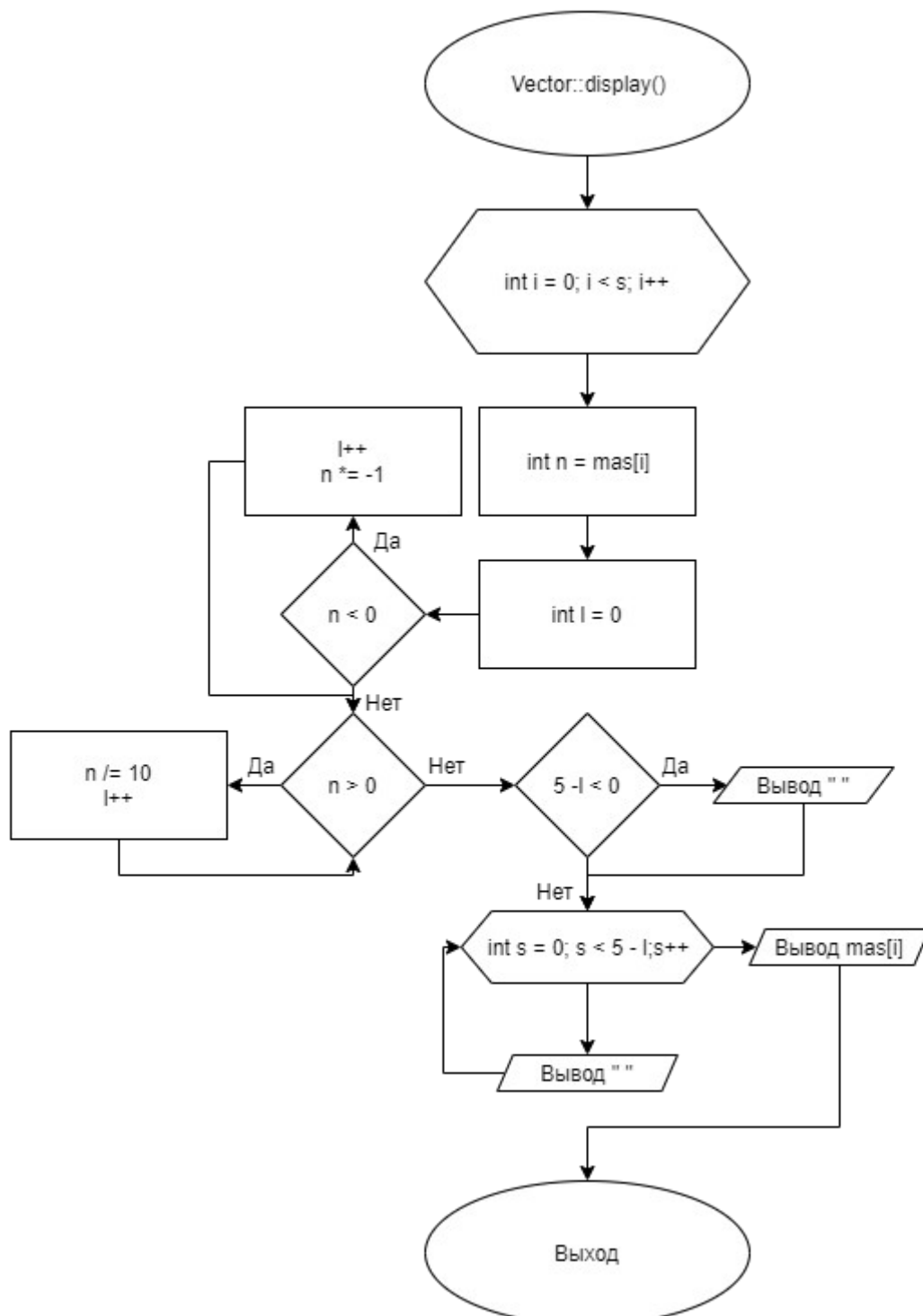


Рис. 2. Блок-схема алгоритма.

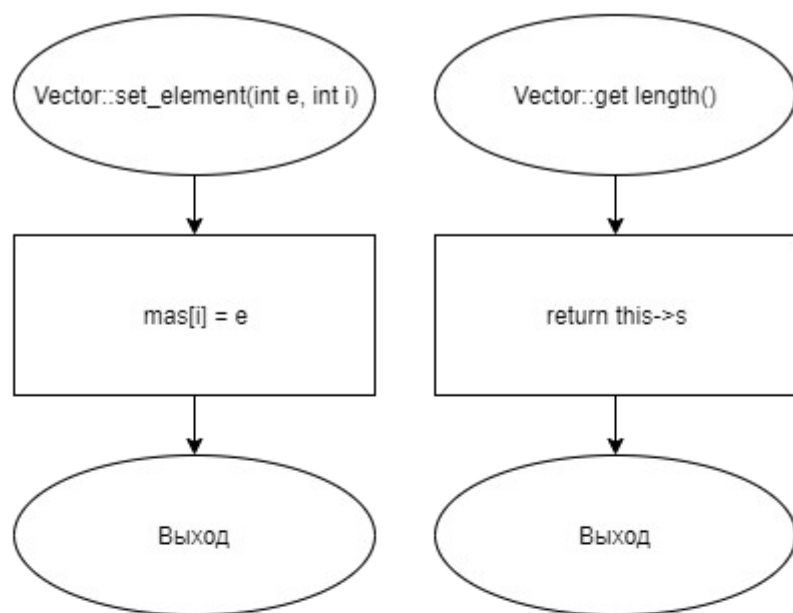


Рис. 3. Блок-схема алгоритма.

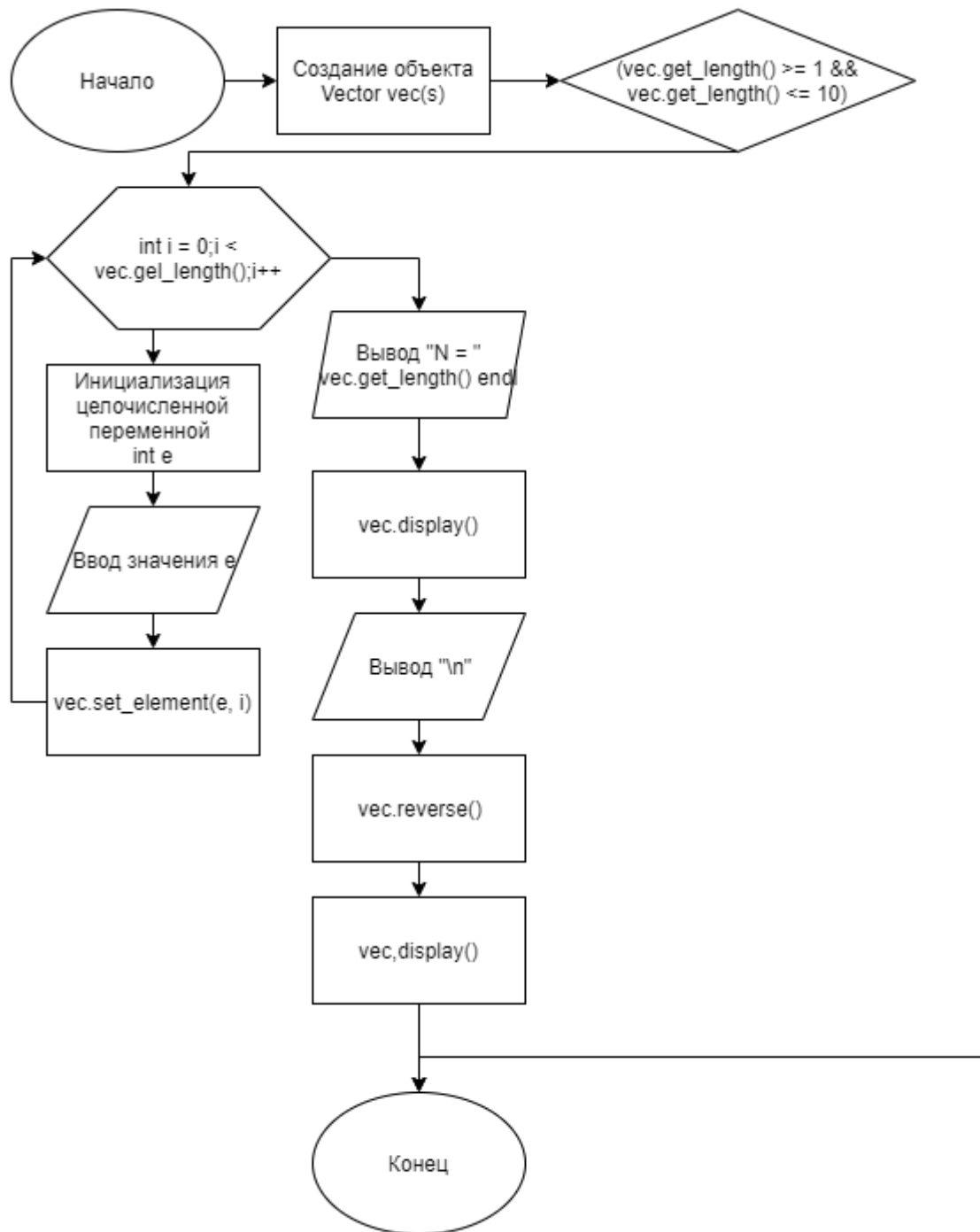


Рис. 4. Блок-схема алгоритма.

Код программы

Программная реализация алгоритмов для решения задачи представлена ниже.

Файл main.cpp

```
#include <iostream>
#include "Vector.h"

using namespace std;

int main()
{
    Vector vec;
    if((vec.get_length() >= 1 && vec.get_length() <= 10)){
        for(int i = 0; i < vec.get_length(); i++){
            int e;
            cin >> e;
            vec.set_element(e, i);
        }
        cout << "N = " << vec.get_length() << endl;
        vec.display();
        cout << "\n";
        vec.reverse();
        vec.display();
        return(0);
    }
}
```

Файл Vector.cpp

```
#include <iostream>
#include "Vector.h"

using namespace std;

Vector::Vector(){
    int len;
    cin >> len;
    if(!(len >= 1 && len <= 10)){
        cout << "Invalid input";
    }else{
        this->s = len;
        this->mas = new int[len];
    }
}
```

```

}

Vector::~Vector(){
    delete [] this->mas;
}

void Vector::reverse(){
    for(int i = 0; i < this->s / 2; i++){
        int buf = this->mas[this->s - i - 1];
        this->mas[this->s - i - 1] = this->mas[i];
        this->mas[i] = buf;
    }
}

void Vector::display(){
    for(int i = 0; i < this->s; i++){
        int n = this->mas[i];
        int l = 0;
        if(n <= 0){
            l++;
            n *= -1;
        }
        while(n > 0){
            n /= 10;
            l++;
        }
        if(5 - l < 0)cout << " ";
        else{
            for(int s = 0; s < 5 - l; s++){
                cout << " ";
            }
        }
        cout << this->mas[i];
    }
}

void Vector::set_element(int e, int i){
    this->mas[i] = e;
}

int Vector::get_length(){
    return this->s;
}

```

Файл Vector.h

```

#ifndef Vector_h
#define Vector_h

class Vector{
private:
    int s;
    int *mas;
public:

```

```
    Vector();  
    ~Vector();  
    void reverse();  
    void display();  
    void set_element(int e, int i);  
    int get_length();  
};  
  
#endif
```

Тестирование

Результат тестирования программы представлен в следующей таблице.

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
0	Invalid input	Invalid input
-10	Invalid input	Invalid input
2 1 2	N = 2 1 2 2 1	N = 2 1 2 2 1
4 1 2 3 4	N = 4 1 2 3 4 4 3 2 1	N = 4 1 2 3 4 4 3 2 1
1 1	N = 1 1 1	N = 1 1 1
10 4 3 2 1 0 -1 -2 -3 -4 -5	N = 10 4 3 2 1 0 -1 -2 -3 -4 -5 -5 -4 -3 -2 -1 0 1 2 3 4	N = 10 4 3 2 1 0 -1 -2 -3 -4 -5 -5 -4 -3 -2 -1 0 1 2 3 4
6 1 12 123 1234 12345 123456	N = 6 1 12 123 123412345 123456 12345612345 1234 123 12 1	N = 6 1 12 123 123412345 123456 12345612345 1234 123 12 1
7 0 1 -1 12 123 1234 1234567	N = 7 0 1 -1 12 123 1234 1234567 1234567 1234 123 12 -1 1 0	N = 7 0 1 -1 12 123 1234 1234567 1234567 1234 123 12 -1 1 0

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ (ИСТОЧНИКОВ)

1. Васильев А.Н. Объектно-ориентированное программирование на C++. Издательство: Наука и Техника. Санкт-Петербург, 2016г. 543 стр.
2. Шилдт Г. C++: базовый курс. 3-е изд. Пер. с англ.. — М.: Вильямс, 2017. — 624 с.
3. Методическое пособие для проведения практических заданий, контрольных и курсовых работ по дисциплине «Объектно-ориентированное программирование» [Электронный ресурс] – URL: https://mirea.aco-avrrora.ru/student/files/methodichescoe_posobie_dlya_laboratornyh_rabot_3.pdf (дата обращения 05.05.2021).
4. Приложение к методическому пособию студента по выполнению заданий в рамках курса «Объектно-ориентированное программирование» [Электронный ресурс]. URL: https://mirea.aco-avrrora.ru/student/files/Prilozheniye_k_methodichke.pdf (дата обращения 05.05.2021).
5. Видео лекции по курсу «Объектно-ориентированное программирование» [Электронный ресурс]. АСО «Аврора».
6. Антик М.И. Дискретная математика [Электронный ресурс]: Учебное пособие /Антик М.И., Казанцева Л.В. — М.: МИРЭА — Российский технологический университет, 2018 — 1 электрон. опт. диск (CD-ROM).