

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ФАКУЛЬТЕТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК
ИМЕНИ ПРОФЕССОРА Н.И.ЧЕРВЯКОВА**

ЛАБОРАТОРНАЯ РАБОТА №18

Алгоритмизация и программирование

Вектор

Выполнил студент:

Сивко Иван Андреевич

студент 2 курса

группа ПМИ-б-о-23-2,

направление подготовки 01.03.02

Проверил:

Ассистент кафедры вычислительной
математики и кибернетики, к.ф.-м.н.,

Черкашина Анастасия Андреевна

Вариант 9

Цель:

- Совершенствование навыков разработки программ в среде программирования MS VStudio
- Совершенствование навыков в программировании с использованием векторов
- Исследование процесса формирования вектора
- Исследование операций с элементами векторов

Задание 1

Используя полученную при выполнении лабораторной работы 10 в

1 Условие:

Используя полученную при выполнении лабораторной работы 10 в Задании II программу, реализовать возможность сохранения и обработки данных с использованием вектора.

Условие задания 2 лабораторной 10:

В одномерном массиве, состоящем из n вещественных элементов, вычислить:

- максимальный по модулю элемент массива;
- преобразовать массив таким образом, чтобы элементы, равные нулю, располагались после всех остальных.

2 Алгоритм / Мат. модель

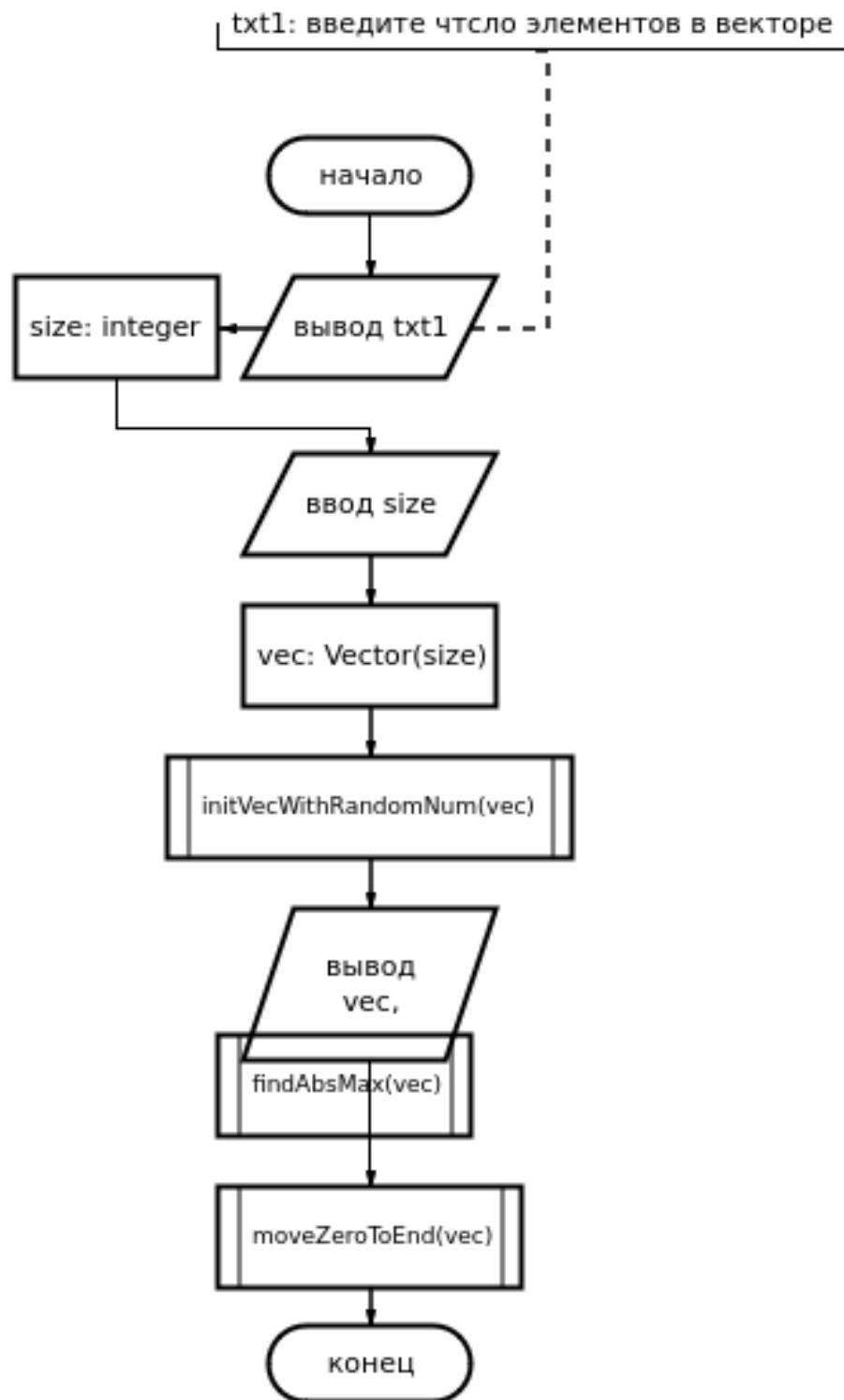
Программа инициализирует а затем заполняет вектор размера введенного пользователем числами от -100 до 100 затем выводит этот вектор, после выводит максимальный элемент массива и преобразует его таким образом, чтобы элементы, равные нулю, располагались после всех остальных.

1. ввод размера для вектора `vec` (в переменную `size` типа `size_t` (*aka unsigned long*))
2. инициализация вектора `vec` типа `double`
3. Заполнение вектора случайными значениями в пределах от -100 до 100
4. вывод вектора `vec` после заполнения и вывод максимального по абсолютной величине значения вектора
5. перемещение всех 0 в `vec` в конец вектора
6. вывод вектора `vec` после перемещения всех 0 в конец вектора

Название	Тип	Описание
Классы и структуры		
Randgen	class	Генератор случайных чисел с различными распределениями.
Переменные-члены Randgen		
state	std::mt19937	Генератор случайных чисел Mersenne Twister, инициализированный функцией helpInitMt.
Функции-члены Randgen		
helpInitMt()	static std::mt19937	Инициализирует генератор случайных чисел с использованием текущего времени и случайных значений от std::random_device.
get<T>(const T&, const T&)	static T	Генерирует случайное число типа T в указанном диапазоне (целое или вещественное).
Другие функции		
abs(T n)	constexpr T	Возвращает модуль числа n.
swap(T& a, T& b)	void	Обменивает значения переменных a и b.
operator«	std::ostream&	Перегрузка оператора « для вывода элементов вектора std::vector<T> в поток.
initVecWithRandomNum	void	Инициализирует вектор случайными числами (используя функцию генерации случайных чисел, по умолчанию Randgen::get).
findAbsMax	const T&	Находит и возвращает элемент вектора с наибольшим абсолютным значением.
moveZerosToTheEnd	void	Перемещает все нулевые элементы вектора в конец, сохраняя порядок остальных элементов.
Переменные main		
size	size_t	Количество элементов вектора, вводимое пользователем.
vec	std::vector<double>	Вектор вещественных чисел, инициализируемый случайными значениями.

Таблица 1: Переменные функции и классы используемы при решении задачи

3 Диаграмма:



4 Код:

```
#include <cmath>
#include <chrono>
#include <random>
#include <type_traits>
```

```

#include <vector>
#include <iostream>

class Randgen {
    static std::mt19937 helpInitMt() {
        std::random_device rd{};
        std::seed_seq seedSeq {
            static_cast<std::seed_seq::result_type>(
                std::chrono::steady_clock::now().time_since_epoch().count(),
                rd(), rd(), rd(), rd(), rd(), rd(), rd()) };
        return std::mt19937{ seedSeq };
    }
    inline static std::mt19937 state{ helpInitMt() };
public:
    template <typename T>
    using RandomFunc = T(*) (const T&, const T&);

    template <typename T>
    static T get(const T& min, const T& max) {
        if constexpr (std::is_integral<T>::value) {
            return std::uniform_int_distribution<T>(min, max)(state);
        } else if constexpr (std::is_floating_point<T>::value) {
            return std::uniform_real_distribution<T>(min, max)(state);
        } else {
            static_assert(std::is_arithmetic<T>::value, "Unsupported type for Randg");
        }
    }
};

template <typename T>
constexpr T abs(T n) {
    return n<0 ? -n : n;
}

template <typename T>
void swap(T& a, T& b) {
    T temp{a};
    a = b;
    b = temp;
}

template <typename T>
std::ostream& operator<<(std::ostream& out, const std::vector<T>& vec) {
    bool isFirst{true};
    for (double el : vec) {
        out << (isFirst?"":", ") /*<< std::setprecision(3)*/ << el;
        isFirst = false;
    }
    return out;
}

template <typename T>

```

```

void initVecWithRandomNum(std::vector<T>& vec, Randgen::RandomFunc<T> func=Randgen:
    for (double& el : vec) {
        el = trunc(
            func(0., 2.)>1 ? func(-100., 100) : 0
        );
    }
}

template <typename T>
const T& findAbsMax(const std::vector<T>& vec) {
    const T *absMax{&vec[0]};
    for (size_t i{1}; i<vec.size(); ++i)
        if (abs(vec[i]) > abs(*absMax))
            absMax = &vec[i];
    return *absMax;
}

template <typename T>
void moveZerosToTheEnd(std::vector<T>& vec) {
    size_t lnz{ vec.size() - 1};
    for (size_t i = 0; i < lnz; ++i) {
        if (vec[i] == 0) {
            while (lnz > i && vec[lnz] == 0)
                --lnz;
            std::swap(vec[i], vec[lnz]);
        }
    }
}

int main(int32_t, const char**) {
    std::cout << "Enter number of elements in arr: ";
    size_t size;
    std::cin >> size;
    std::vector<double> vec(size);
    initVecWithRandomNum(vec);
    std::cout << vec << '\n' << findAbsMax(vec) << '\n';
    moveZerosToTheEnd(vec);
    std::cout << vec << '\n';
    return EXIT_SUCCESS;
}

```

source code

5 Результат работы программы:

```
[john@arch data]$ ./a.out
Enter number of elements in arr: 8
65, 7, 0, 0, 33, 0, 96, 0
96
65, 7, 96, 33, 0, 0, 0, 0
[john@arch data]$ ./a.out
Enter number of elements in arr: 8
-29, 0, 0, 0, 24, 0, 0, -51
-51
-29, -51, 24, 0, 0, 0, 0, 0
[john@arch data]$ ./a.out
Enter number of elements in arr: 10
0, 0, 0, -40, 0, 0, 0, 0, 0, -46
-46
-46, -40, 0, 0, 0, 0, 0, 0, 0, 0
```

Задание 2

1 Условие:

1. Разработать соответствующую варианту программу с использованием имеющегося описания, отладить ее и перерешать с использованием векторов.
2. Подготовить набор тестов, подтверждающих правильность работы программы.
3. Оформить отчет, включив в него постановку задачи коды программ и результаты их работы.

Задано описание:

```
typedef float* Vector[100];
Vector x;
```

Считая, что все элементы вектора *x* отличны от *NULL*, описать

- процедуру *del(x)*, которая в векторе *x* все ссылки указывающие на повторяющиеся элементы заменяет значением *NULL*;
- процедуру *Inp(x)* – формирования вектора *x*;
- процедуру *Out(x)* – вывода чисел, на которые ссылаются элементы вектора *x* отличные от *NULL*.

2 Алгоритм / Мат. модель