Федеральное агентство по образованию Российской Федерации

Государственное образовательное учреждение

высшего профессионального образования

Нижегородский государственный университет им. Н. И. Лобачевского

Институт информационных технологий математики и механики

Отчёт по лабораторной работе

Массивы и методы их сортировки

Выполнила:

студент ф-та ИИТММ гр. 3821Б1ПМ3

Виденева Е. В.

Проверил:

Заведующий лабораторией

суперкомпьютерных технологий и

высокопроизводительных

вычислений

Лебедев И. Г.

Нижний Новгород

2021 г.

Оглавление

[**Введение** 3](#_Toc89628578)

[**1.** **Постановка задачи** 4](#_Toc89628579)

[**2.** **Руководство пользователя** 5](#_Toc89628580)

[**3.** **Руководство программиста** 9](#_Toc89628581)

[**a.** **Описание структуры кода** 9](#_Toc89628582)

[**b.** **Описание структуры данных** 14](#_Toc89628583)

[**c.** **Описание алгоритмов** 14](#_Toc89628584)

[**4.** **Эксперименты** 21](#_Toc89628585)

[**Заключение** 23](#_Toc89628586)

[**Литература** 24](#_Toc89628587)

# **Введение**

Необходимость отсортировать какие–либо величины возникает в программировании очень часто. К примеру, входные данные подаются «вперемешку», а вашей программе удобнее обрабатывать упорядоченную последовательность. Существуют ситуации, когда предварительная сортировка данных позволяет сократить содержательную часть алгоритма в разы, а время его работы — в десятки раз. Сортировка массивов не исключение. Использование массивов дает возможность хранить связанные значения одного и того же типа. При работе с массивами может возникнуть необходимость сортировки значений массива как по возрастанию (от меньшего к большему), так и по убыванию (от большего к меньшему). Для сортировки массива есть несколько различных алгоритмов: метод пузырька, метод выбора, метод Шелла и быстрая сортировка. В данной программе мы рассмотрим три метода сортировки.

1. **Постановка задачи**

Реализовать сортировки массивов данных (тип данных определяется преподавателем) задаваемых: обязательно случайно, дополнительно с клавиатуры или из файла.

Реализовать сортировки: пузырьком, вставкой, быстрая.

Сравнить время работы, сделать выводы.

Первая программа создает текстовый файл с записанными в него числами. Программа принимает количество чисел n, максимальное и минимальное значение.

Вторая программа читает текстовый файл с набором чисел, выводит консольный интерфейс (печать, сортировка, сброс, выход), выполняет выбранные действия.

1. **Руководство пользователя**

При запуске первой части под названием Project 0101 программа запросит ввести с клавиатуры N (количество всех чисел), max (верхняя граница диапазона чисел) и min (нижняя граница диапазона чисел).

Затем программа выведет массив, удовлетворяющий введенным с клавиатуры условиям. (см. рис. 1)

Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст, устройство, счетчик

Автоматически созданное описание

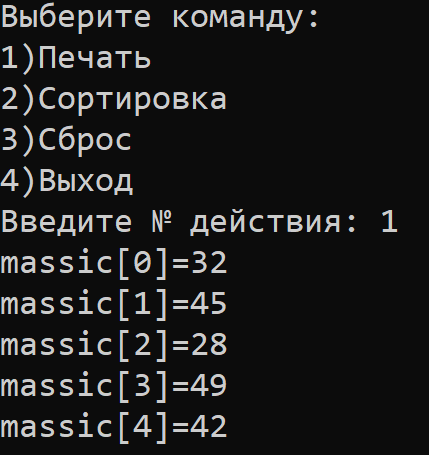
Рисунок 1 Вывод массива

Рисунок 2 Вывод в случае ошибки

ВАЖНО! Если вы введете такие числа, что N<0, max< min или массив=0 программа выведет на экран ОШИБКА (причина ошибки):( (См. рис. 2)

Далее необходимо перейти ко второй части программы под названием Progect0102. При ее запуске вы увидите текст "Выберите команду 1)Печать 2)Сортировка 3)Сброс 4)Выход"

"Введите № действия:” Далее необходимо ввести с клавиатуры цифру (от 1 до 4) команды, которую вы хотите выполнить.(См. рисунок 3)

* При вводе 1 программа напечатает не отсортированный массив, который ранее был напечатан первой частью программы.( См. рис. 3.1)
* При вводе 2 программа выведет текст "Выберите метод сортировки: 1.Пузырьком 2. Вставкой 3. Быстрая". Для выбора необходимой вам сортировки введите цифру от 1 до 3.(см. рис. 3.2)(При вводе цифры больше 3, программа выдаст ошибку(См. рис. 3.2’)
* При вводе 3 массив будет сброшен к исходному. (См. рис. 3.3)
* При вводе 4 программа завершит работу. (см. рис. 3.4)
* Изображение выглядит как текст

  Автоматически созданное описаниеПри вводе числа большего 4 программа выдаст ошибку. (см. рис. 3.5)

Рис. 3 Список команд

Рис. 3.1 Команда 1

Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание

Рис. 3.2 Команда 2

Рис. 3.3 Команда 3

Изображение выглядит как текст

Автоматически созданное описание

Рис. 3.2’

Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст, знак

Автоматически созданное описание

Рис. 3.4 Команда 4

Рис. 3.5 Некорректный ввод команды

Так же во время выполнения каждой из сортировок программа замеряет время работы и показывает его пользователю:

1. **Руководство программиста**
2. **Описание структуры кода**

Первая программа Project0101

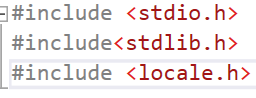
1. Подключение библиотек, с которыми предстоит работать (См. рис 4.1)

Рис. 4.1 Библиотеки

1. Объявление функции main( ), открытие файла получение данных с клавиатуры и записывание в файл.(см. рис. 4.2)

Изображение выглядит как текст

Автоматически созданное описание

Рис. 4.2 Функция main()

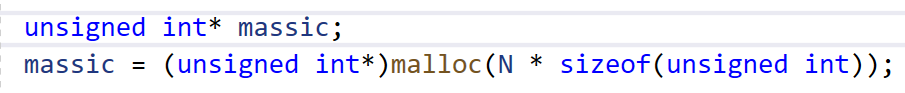
1. Проверка полученных данных на корректность. (см. рис. 4.3)
2. Объявление массива типа unsigned int, присвоение ему случайный чисел и его запись в фай. (см. рис. 4.4 и 4.4’)

Рис. 4.4 Объявление массива

Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описаниеРис. 4.3 Рис.4.4’ Присвоение случайных чисел массиву

1. Очистка памяти и вывод на экран массива.(см. рис. 4.5)

Изображение выглядит как текст

Автоматически созданное описание

Рис.4. 5 Вывод массива

Вторая программа Progect0102

1. Изображение выглядит как текст

   Автоматически созданное описаниеПодключение библиотек, с которыми предстоит работать. (см. рис. 5.1)

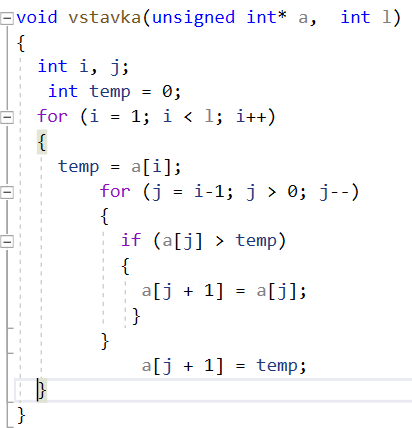
Рис.5.1 Библиотеки

1. Изображение выглядит как текст

   Автоматически созданное описаниеИзображение выглядит как текст

   Автоматически созданное описаниеСоздание дополнительной функции для вывода массива на экран (см. рис. 5.2)

Рис. 5.2 Функция вывода массива

Изображение выглядит как текст

Автоматически созданное описание Рис. С Сортировка «Быстрая»

Рис. А Сортировка «Пузырьком»

Рис. Б Сортировка «Вставкой»

1. Создание дополнительных функций для сортировки массива. (см. рисунки А, Б, С)
2. Изображение выглядит как текст

   Автоматически созданное описаниеОбъявление переменных и открытие файла для считывания. (см. рис.5.3)

Рис.5.3

1. Изображение выглядит как текст

   Автоматически созданное описаниеОбъявление массива и его считывание из файла.(см. рис. 5.4)

Рис. 5.4

1. Интерфейс главного меню и его реализация.

Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание Рис. 5.5

1. Очистка памяти и завершение работы. (Рис. 5.6)

Изображение выглядит как текст

Автоматически созданное описание

Рис. 5.6

1. **Описание структуры данных**

Первая программа

1. В программе используется счетчик типа int (i), а так же количество генерируемых чисел N типа int.
2. Переменные min, max, относятся к типу int.
3. Случайные числа random( ) записываются в массив massic типа unsigned int
4. Для функционирования программы требуются такие библиотеки, как stdio.h, stdlib.h, locale.h.
5. File-файл, в который записывается количество случайно генерируемых чисел и сами числа.

Вторая программа

1. В программе используются переменные счетчики типа int (i, j), а так же количество генерируемых чисел N типа int.
2. Переменные a, b относятся к типу double.
3. Временная перменная temp типа int.
4. Случайные числа random( ) записываются в массив massic типа unsigned int
5. File-файл, с которого считывается количество случайно генерируемых чисел и сами числа.
6. Для функционирования программы требуются такие библиотеки, как stdio.h, stdlib.h, locale.h, time.h.
7. **Описание алгоритмов**
8. Алгоритм заполнение массива случайными числами (См. Блок-схему1)

Блок-схема1:

Начало

i=0

Ввод N с клавиатуры

false

massic[i] =rand()-случайное число

true

i<N

Конец цикла

1. Алгоритм сортировки пузырьком (см. Блок-схему 2)

Блок-схема 2:

Начало

Ввод a[], l

i<N

true

false

Конец

j<N-i-1

i++

false

true

a[j]>a[j+1]

true

temp=a[j]

a[j]=a[j+1]

a[j+1]=temp

j++

false

1. Алгоритм быстрой сортировки(см. Блок-схему 3)

Блок-схема 3:

Начало

mas[], l

do

mas[i]<x

i++

true

false

mas[j]>x

j--

false

true

i<=j

temp=mas[i]

mas[i]=mas[j]

mas[j]=temp

true

j>0

false

mas, j+1

true

false

i<u

конец

false

&mas[i], l-1

true



1. Алгоритм сортировки вставкой (см. Блок-схему 4)

Блок-схема 4:

Начало

i<l

конец

false

temp=a[i]

true

j=i-1

(j>0)&&(a[j]>temp

a[j+1]=a[j]

j--

a[i]=temp

true

i++

false

1. **Эксперименты**

Для того, чтобы убедиться в работоспособности программы протестируем ее при вводе различных переменных.

1. Проверка на некорректные данные в первой программе.

Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описаниеПри вводе некорректных данных, при которых N<0 (см. рис. 6.1), max<min(см. рис. 6.2), massic=0(см. рис. 6.3) программа покажет на экране конкретную причину ошибки.

Рис. 6.3

Рис. 6.2

Рис. 6.1

1. Проверка на корректность второй программы

Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описаниеПри вводе в 1 программу корректных чисел (например N=10, max=99, min=6), переходим во вторую программу и проверяем на корректность три вида сортировки (см.рис. 7)

Рис. 7 Вывод 3 сортировок

1. Корректность времени работы сортировок.

Сгенерировав достаточно большое количество случайных чисел\. Получим время работы каждой сортировки. Для упрощения понимания результатов используем таблицу.

|  |  |  |
| --- | --- | --- |
| Тип сортировки | Размер исходных данных | Время работы, мс |
| Пузырек | 10000 | 168 |
| Вставка | 10000 | 72 |
| Быстрая | 10000 | 0 |

Таким образом самой быстрой сортировкой для 10 000 элементов оказалась быстрая сортировка

# **Заключение**

Для выполнения лабораторной работы №2 мною были написаны две исправно работающие программы на языке программирования Си, которые выполняют конкретные задачи, поставленные преподавателем.

Благодаря выполнению данной лабораторной работы я смогла узнать больше информации про работу с массивами. Я подробно изучила три вида сортировок массивов, научилась создавать файлы, записывать и считывать данные из этого файла, а также поработала с определением времени выполнения конкретной операции.

Данная работа помогла мне узнать новую информацию, а именно какая из сортировок быстрее работает. Я выяснила, что самой интенсивной из всех сортировок являются быстрая сортировка.

# **Литература**

1. Б.В. Керниган, Д.М. Ричи.//ЯЗЫК Срежим доступа: [kr.pdf](file:///C:\Users\Lenovo\Downloads\kr.pdf)
2. M. УЭИТ С. ПРАТА Д. МАРТИН //ЯЗЫК С для начинающихрежим доступа:

[<DFC7DBCA20D1C8A0A0A0A04D2E20D3DDC9D22C20D12E20CFD0C0D2C02C20C42E20CCC0D0D2C8CD> (ded32.net.ru)](http://storage.ded32.net.ru/Lib/TX/CPrimerPlus.pdf)