|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |   Институт информационных технологий |
| Кафедра вычислительной техники |
|  |

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №11** | |
| **по дисциплине** | |
| **«Алгоритмические основы обработки данных»** | |
|  | |
| Выполнил студент группы ИВБО-11-23  *(учебная группа)* | Туктаров Т.А. |
| Принял старший преподаватель | Асадова Ю.С. |

|  |  |  |
| --- | --- | --- |
| Практическая работа выполнена | «6» ноября 2024г. | *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*  *(подпись студента)* |
|  |  |  |
| «Зачтено» | «2» декабря 2024г. | *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*  *(подпись руководителя)* |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт информационных технологий

Кафедра вычислительной техники

Выполнено \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Т.А. Туктаров/

Зачтено \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Ю.С. Асадова/

**Задание на практическую работу №11**

Дисциплина: «Алгоритмические основы обработки данных»

Студент Туктаров Тимур Азатович Шифр 23И0087 Группа ИВБО-11-23

**1. Тема**: «использование библиотечных функций для обработки текста».

**2. Срок сдачи студентом законченной работы:** 27.11.2024.

**3. Исходные данные:** текстовый файл.

**4. Задание:**

Разработать программу, выполняющую следующие функции:

• ввод данных об n объектах из текстового файла в массив структур (0<n<=50);

• сортировку массива структур по возрастанию значений одного из полей структуры;

• вывод данных об объектах на экран в упорядоченном по возрастанию виде;

• поиск объекта по значению одного из полей;

• запись упорядоченных данных об объектах в двоичный файл;

• чтение двоичного файла..

**5. Содержание отчета:**

* титульный лист;
* задание;
* оглавление;
* введение;
* основные разделы отчета;
* заключение;
* список использованных источников;

Руководитель работы Ю.С. Асадова \_\_\_\_\_\_\_\_\_\_\_\_ «6» ноября 2024г.

подпись

Задание принял к исполнению Т.А Туктаров \_\_\_\_\_\_\_\_\_\_\_\_ «6» ноября 2024г.

подпись

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc181801359)

[1 ОСНОВНОЙ РАЗДЕЛ 5](#_Toc181801360)

[2 БЛОК-СХЕМА АЛГОРИТМА.. 6](#_Toc181801361)

[3 ИСХОДНЫЙ КОД 10](#_Toc181801362)

[4 ПРИМЕР РАБОТЫ ПРОГРАММЫ 12](#_Toc181801363)

[ЗАКЛЮЧЕНИЕ 13](#_Toc181801364)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 14](#_Toc181801365)

# ВВЕДЕНИЕ

В данной практической работе требуется применить функции, работу со строками, динамическими массивами и файлами.

Постановка задачи:

Разработать программу, выполняющую следующие функции:

• ввод данных об n объектах из текстового файла в массив структур (0<n<=50);

• сортировку массива структур по возрастанию значений одного из полей структуры;

• вывод данных об объектах на экран в упорядоченном по возрастанию виде;

• поиск объекта по значению одного из полей;

• запись упорядоченных данных об объектах в двоичный файл;

Алгоритмы чтения файла, сортировки, поиска, вывода данных об объектах и записи данных в файл оформить в виде функций. Для поиска элемента в упорядоченном массиве использовать бинарный поиск. Текстовый файл создать с помощью любого текстового редактора.

# 1 ОСНОВНОЙ РАЗДЕЛ

Программа состоит из 7 функций – main, comp, bin\_search, sort\_vec, read\_vec, print\_vec, write\_vec. Также в программе создается структура bill для содержания информации об чеке. Сначала создается вектор из структур, для хранения данных. Затем программа считывает названия файлов для ввода и вывода значений. Далее вызываются функции read\_vec, print\_vec, sort\_vec, print\_vec. После программа считывает номер чека, который надо вывести и выводит . Далее с помощью функции write\_vec мы записываем данные в бинарный файл.

Функция main является основной. В ней мы считываем названия текстого и бинарного файла, номер чека а также вызываем функции чтения и записи в файлы и обработки массива.

Функция comp – служит компаратором для сортировки динамического массива.

Функция bin\_search – Функция бинарного поиска в массиве чеков по номеру. Возвращает индекс найденного чека или -1, если ничего не было найдено.

Функция sort\_vec – сортирует динамический массив используя функцию sort и компаратор comp

Функция read\_vec – считывает информацию о чеках из текстового файла.

Функция print\_vec – выводит содержимое динамического массива чеков.

Функция write\_vec – записывает в бинарный файл значения из динамического массива.

# 2 БЛОК-СХЕМА АЛГОРИТМА..

Представим описание алгоритма в графическом виде на рисунках 2.1 – 2.2

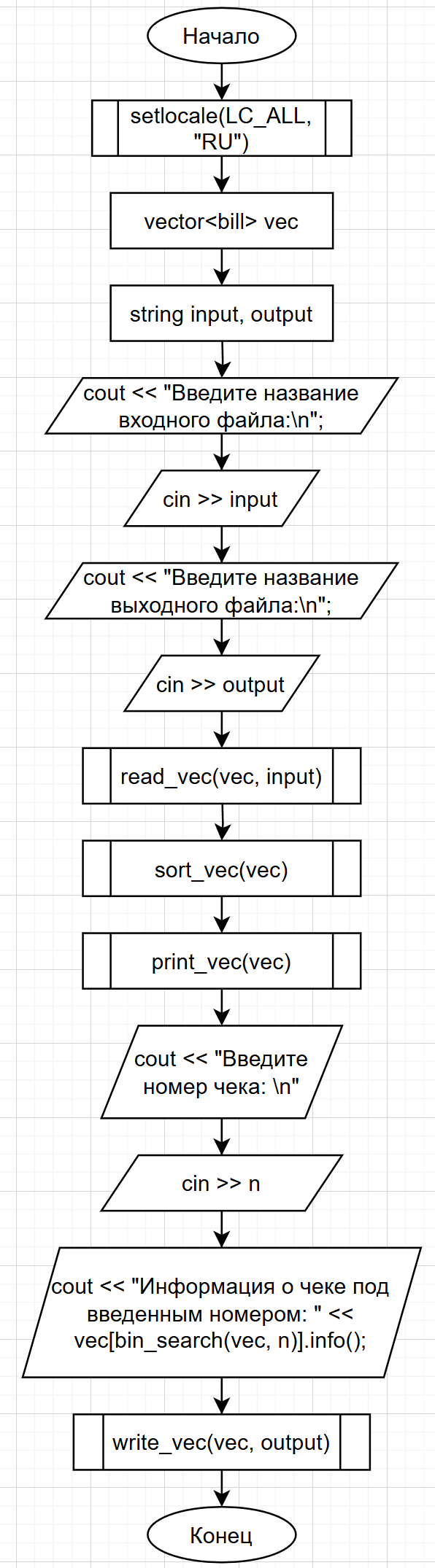


Рисунок 2.1 – Блок – схема алгоритма функции main()

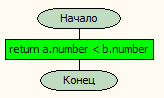


Рисунок 2.2 – Блок-схема алгоритма функции comp()

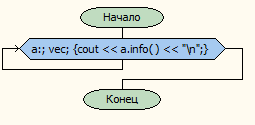


Рисунок 2.3 – Блок-схема алгоритма функции print\_vec()

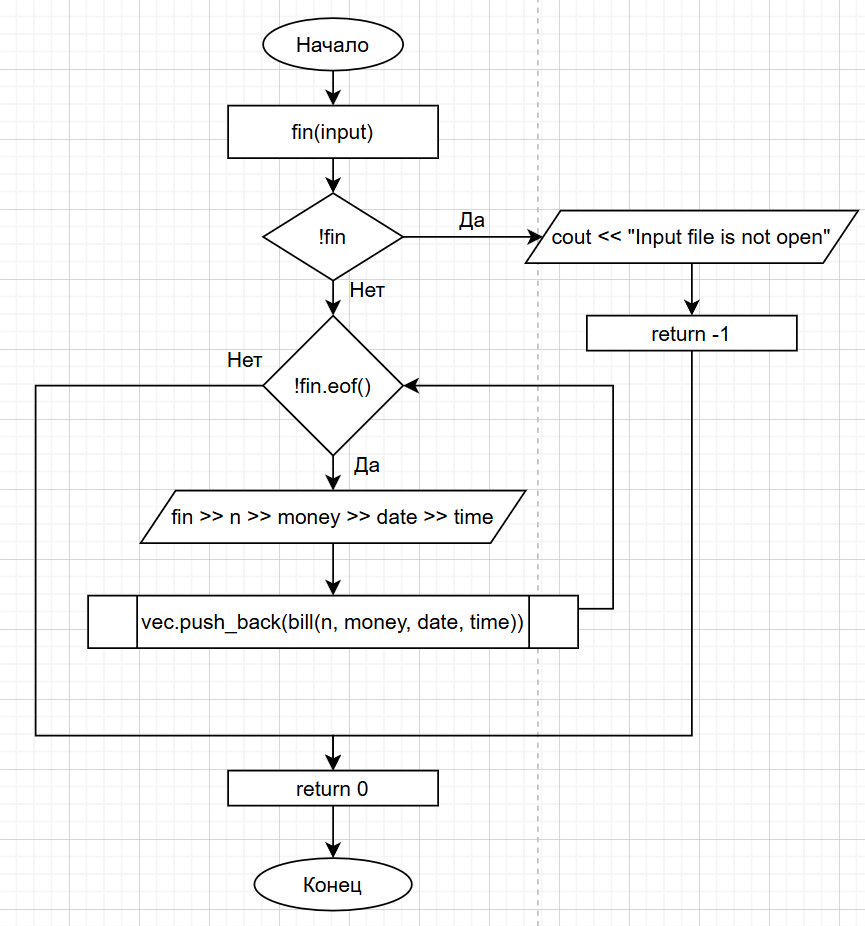


Рисунок 2.4 – Блок-схема алгоритма ф-ции read\_vec()

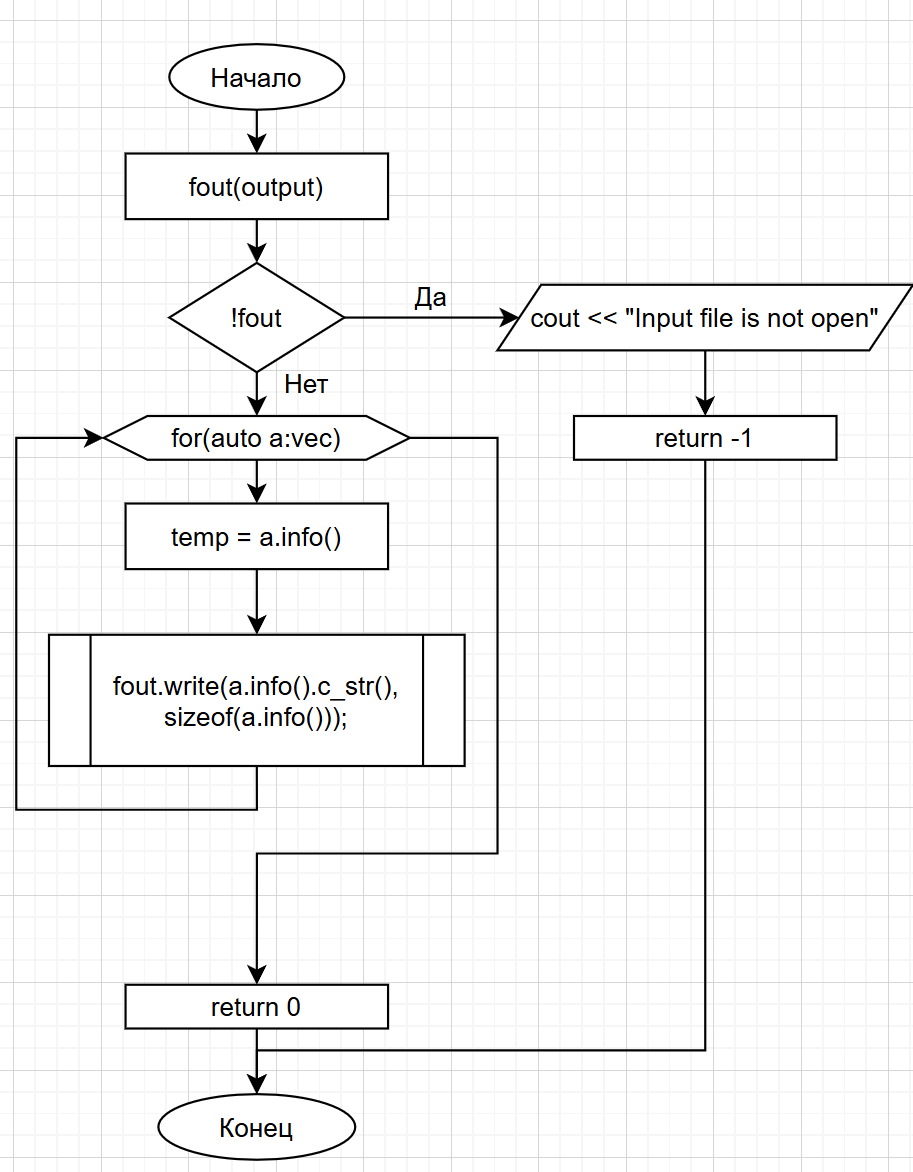


Рисунок 2.5 – Блок-схема алгоритма функции write\_vec()

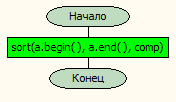


Рисунок 2.6 – Блок-схема алгоритма функции sort\_vec()

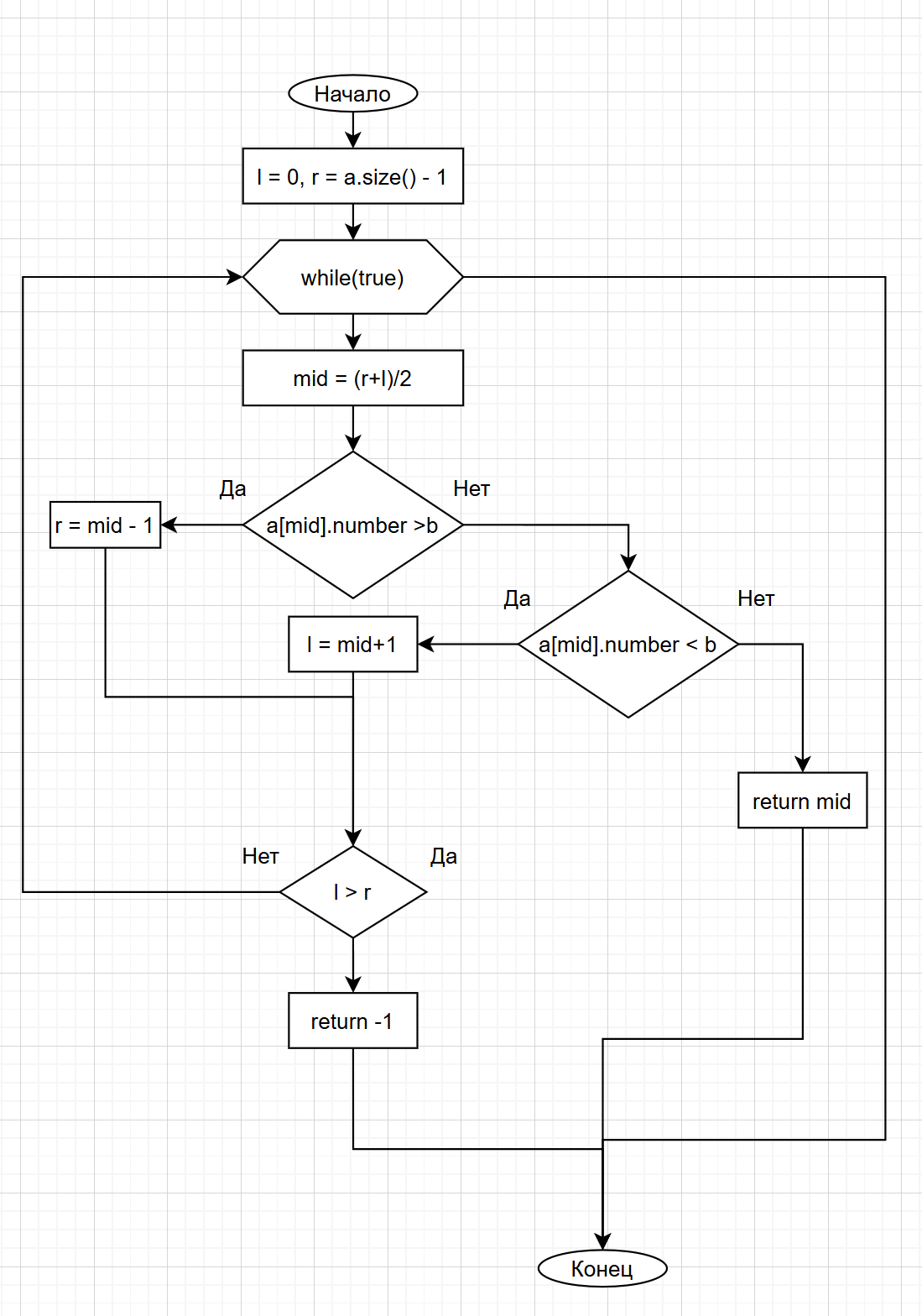


Рисунок 2.7 – Блок-схема алгоритма функции bin\_search

# 3 ИСХОДНЫЙ КОД

Программная реализация алгоритма для решения задачи представлена ниже.

Листинг 3.1 – Исходный код программы

|  |
| --- |
| #include <iostream>  #include <fstream>  #include <string>  #include <vector>  #include <algorithm>  #include <stdio.h>  using namespace std;  struct bill {  int number, money\_sum;  string date, time;  bill(int n, int m, string d, string t) {  number = n;  money\_sum = m;  date = d;  time = t;  }  string info() {  return to\_string(number) + " " + to\_string(money\_sum) + " " + date + " " + time;  }  };  bool comp(bill &a, bill &b) {  return a.number < b.number;  }  int bin\_search(vector<bill>& a, const int& b) {  int l = 0, r = a.size() - 1;  while (1) {  int mid = (r + l) / 2;  //cout << l << " " << r << " " << mid << "\n";  if (a[mid].number > b)r = mid - 1;  else if (a[mid].number < b) l = mid + 1;  else if (a[mid].number == b)return mid;  if (l > r)return -1;  }  }  void sort\_vec(vector<bill>& a) {  sort(a.begin(), a.end(), comp);  }  int read\_vec(vector<bill>& vec, string input) {  ifstream fin(input);  if (!fin) {  cout << "Input file is not open!";  return -1;  }  while (!fin.eof()) {  int n, money;  string date, time;  fin >> n >> money >> date >> time;  vec.push\_back(bill(n, money, date, time));  }  return 0;  }  void print\_vec(vector<bill>& vec) {  for (auto a : vec) {  cout << a.info() << "\n";  }  }  int write\_vec(vector<bill>& vec, string output) {  ofstream fout(output);  if (!fout) {  cout << "Output file is not open!";  return -1;  }  for (auto a : vec) {  string temp = a.info();  fout.write(a.info().c\_str(), sizeof(a.info()));  }  return 0;  }  int main()  {  setlocale(LC\_ALL, "RU");  string input, output;  cout << "Введите название входного файла:\n";  cin >> input;  cout << "Введите название выходного файла:\n";  cin >> output;  vector<bill>vec;  read\_vec(vec, input);  cout << "Содержание файла:\n";  for (int i = 0; i < vec.size(); i++)  print\_vec(vec);  sort\_vec(vec);  cout << "Отсортированные значения:\n";  print\_vec(vec);  cout << "Введите номер чека: \n";  int n;  cin >> n;  cout << "Информация о чеке под введенным номером: " << vec[bin\_search(vec, n)].info();  write\_vec(vec, output);  } |

# 4 Пример работы программы

Пример программы считывающая файл с описанием 3-х чеков, их сортировкой и поиском 2-го чека.



Рисунок 4.1 – Содержание текстового файла.

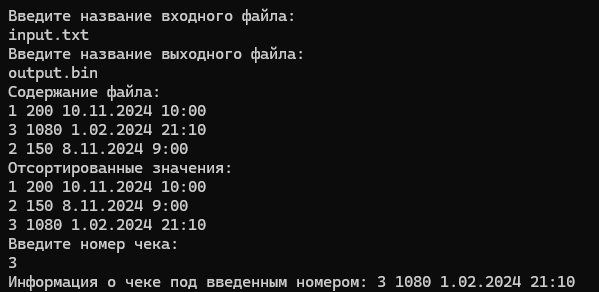


Рисунок 4.2 – Пример работы программы – поиск индекса числа 2.



Рисунок 4.3 – Содержание бинарного файла.

# ЗАКЛЮЧЕНИЕ

В ходе выполнения данной практической работы была реализована программа для работы с файлами. Также были приобретены навыки работы с массивами данных, и обработкой текстовой информации.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Лозовский В.В. Алгоритмические основы обработки данных: учебное пособие / Лозовский В.В., Платонова О.В., Штрекер Е.Н. — М.: МИРЭА – Российский технологический университет, 2022. – 337 с.

2. Платонова О.В. Алгоритмические основы обработки данных: методические указания / Платонова О.В., Асадова Ю.С., Расулов М.М. — М.: МИРЭА – Российский технологический университет, 2022. — 73 с.

3. Белик А.Г. Алгоритмы и структуры данных: учебное пособие / А.Г. Белик, В.Н. Цыганенко. — Омск: ОмГТУ, 2022. — 104 с. — ISBN 978-5-8149-3498-7. — Текст: электронный // Лань: электронно-библиотечная система. — URL: https://e.lanbook.com/book/343688 (дата обращения: 23.09.2024)

4. Павлов Л.А. Структуры и алгоритмы обработки данных / Л.А. Павлов, Н.В. Первова. — 2-е изд., стер. — Санкт-Петербург: Лань, 2022. — 256 с. — ISBN 978-5-507-44105-1. — Текст: электронный // Лань: электронно-библиотечная система. — URL: https://e.lanbook.com/book/207563 (дата обращения: 23.09.2024)

5. Пантелеев Е.Р. Алгоритмы и структуры данных: учебное пособие / Е.Р. Пантелеев, А.Л. Алыкова. — Иваново: ИГЭУ, 2018. — 142 с. — Текст: электронный // Лань: электронно-библиотечная система. — URL: https://e.lanbook.com/book/154576 (дата обращения: 23.09.2024)