



Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и управление»

КАФЕДРА ИУК4 «Программное обеспечение ЭВМ,
информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №8

«Обработка файлов произвольного доступа»

ДИСЦИПЛИНА: «Высокоуровневое программирование»

Выполнил: студент гр. ИУК4-11Б _____ (_____ Суриков Н.С _____)
(подпись) (Ф.И.О.)

Проверил: _____ (_____ Пчелинцева Н.И. _____)
(подпись) (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Цель работы: приобретение практических навыков создания и обработки файлов произвольного доступа.

Задачи:

1. Познакомиться с организацией файлов произвольного доступа.
2. Изучить основные программные средства для произвольной обработки файлов указанного типа.

Вариант 8

Формулировка общего задания:

Задача 1:

Запись имеет вид:

- номер читательского билета
- автор книги
- название
- дата заказа

Создать файл из 10 записей, просмотреть файл, добавить в файл (в начало и конец) новую информацию.

Поменять местами первую и последнюю записи в файле.

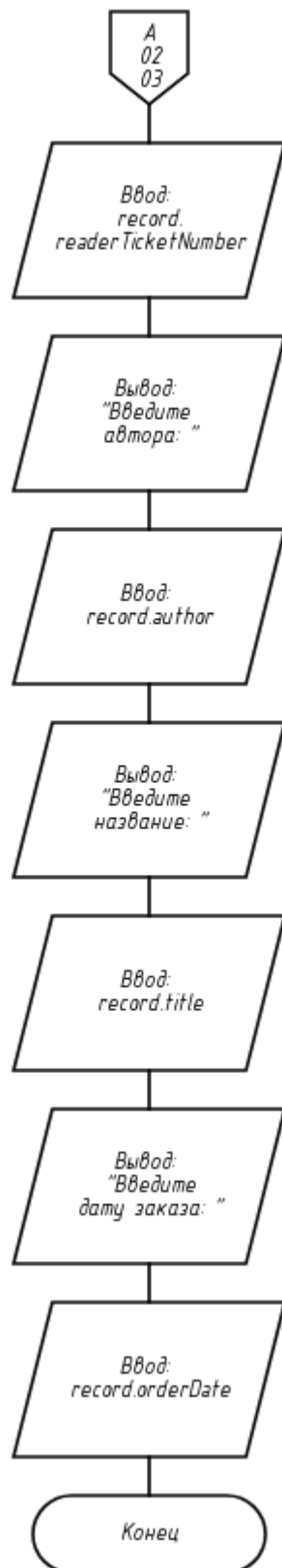
Задача 2:

Для произвольного текстового файла программа составляет файл записей фиксированной длины, содержащий файловые указатели на строки текстового файла. Программа производит логическую перестановку строк, не меняя самого текстового файла.

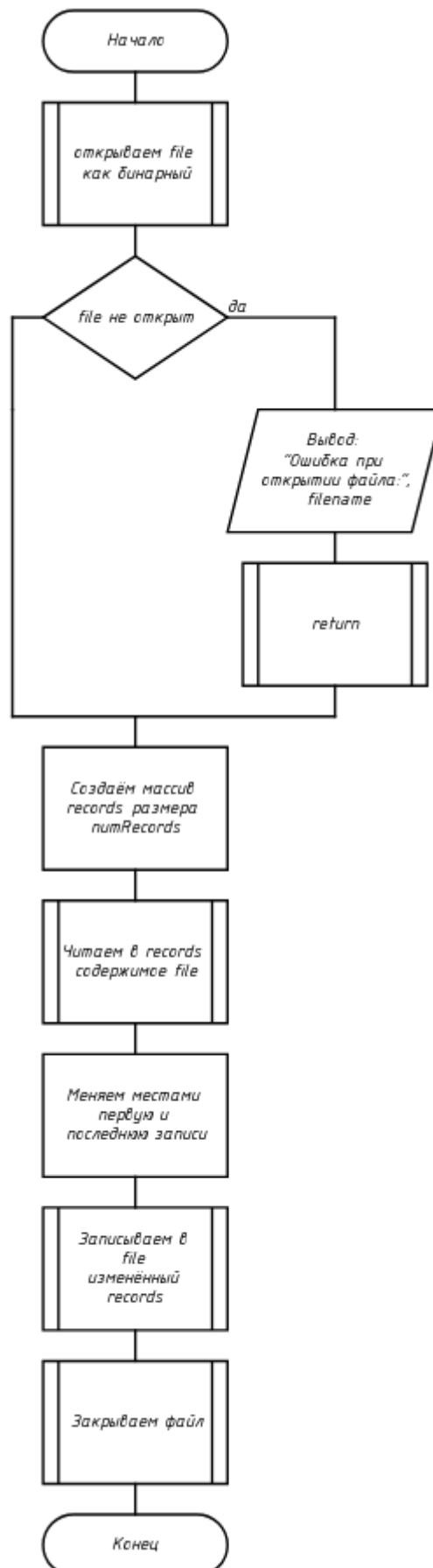
Блок схемы подпрограмм:



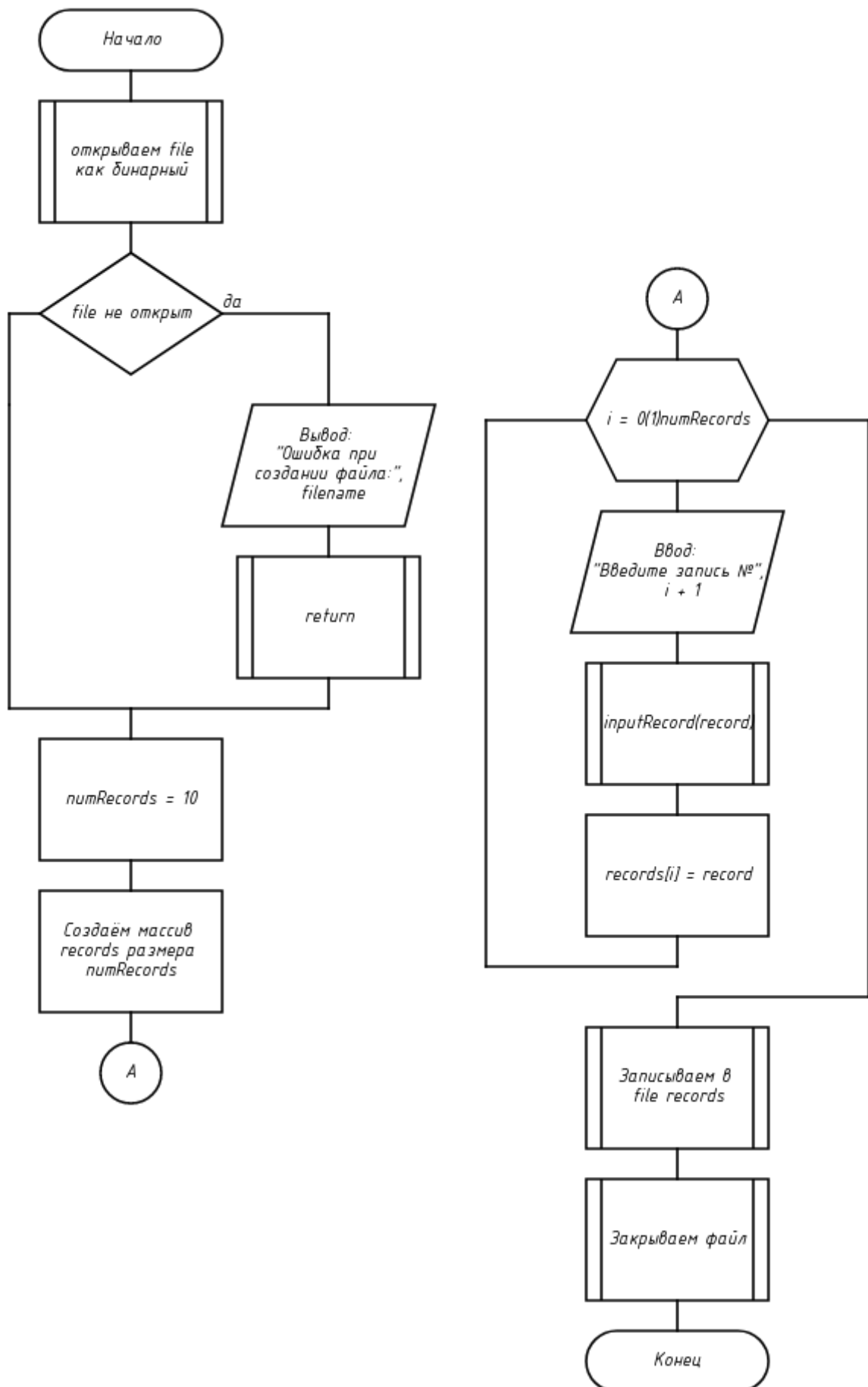
`void inputRecord(Record &record)`



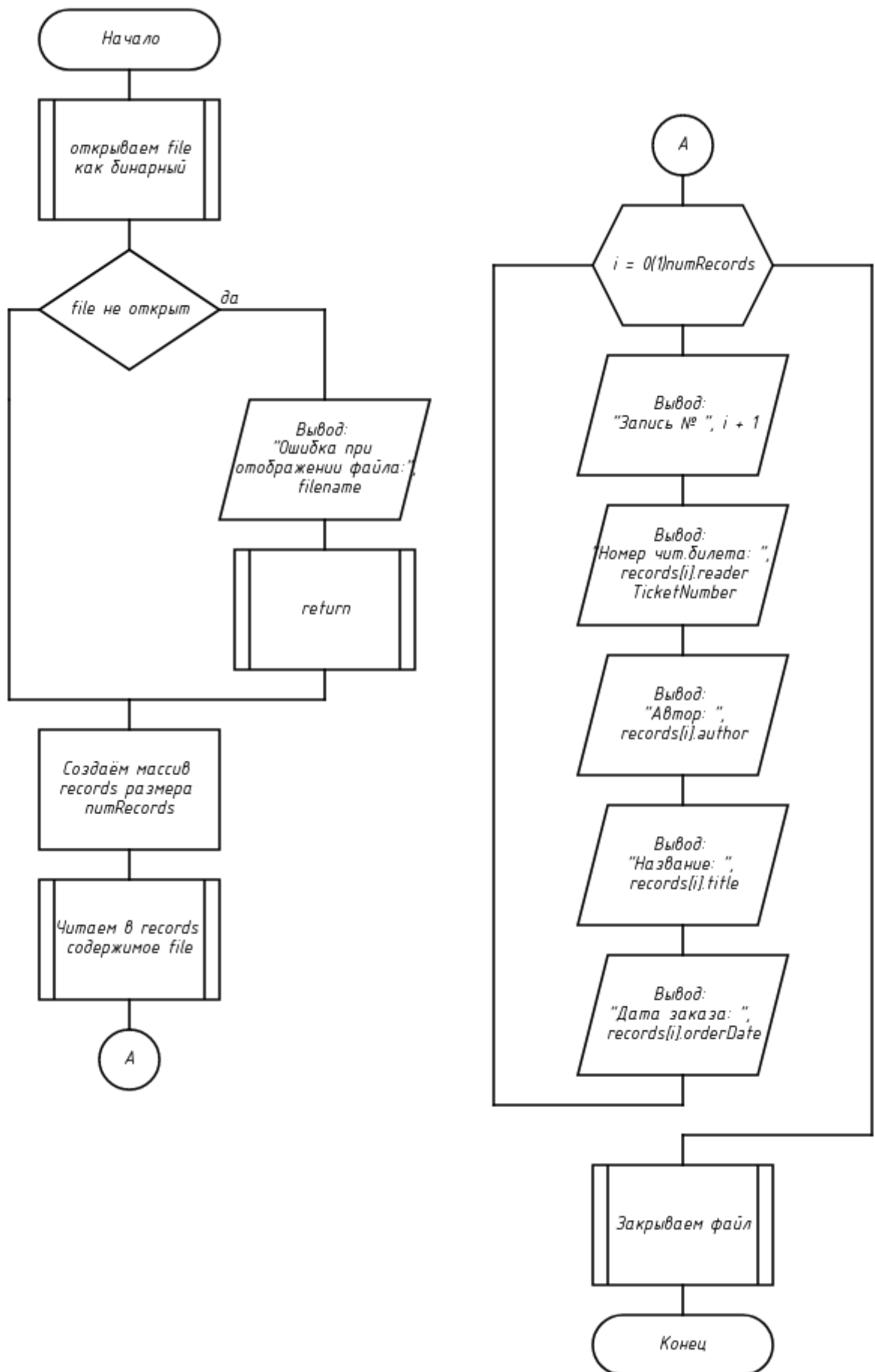
void swapFirstAndLastRecord(const char *filename, int numRecords)



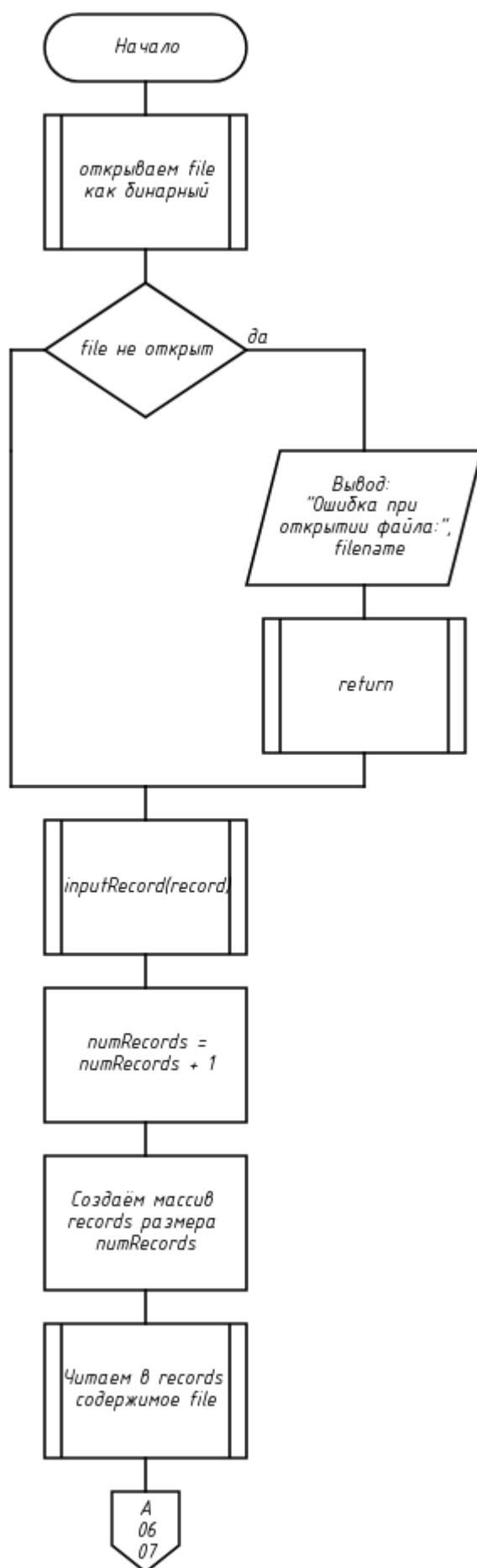
void createFile(const char *filename, int &numRecords, bool)

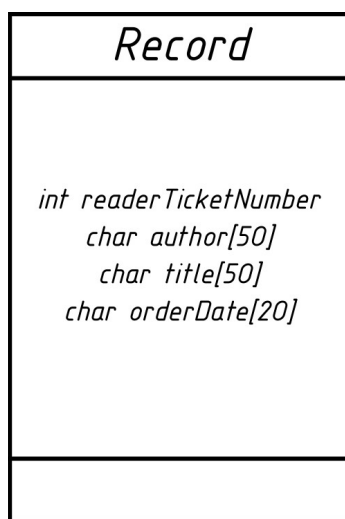
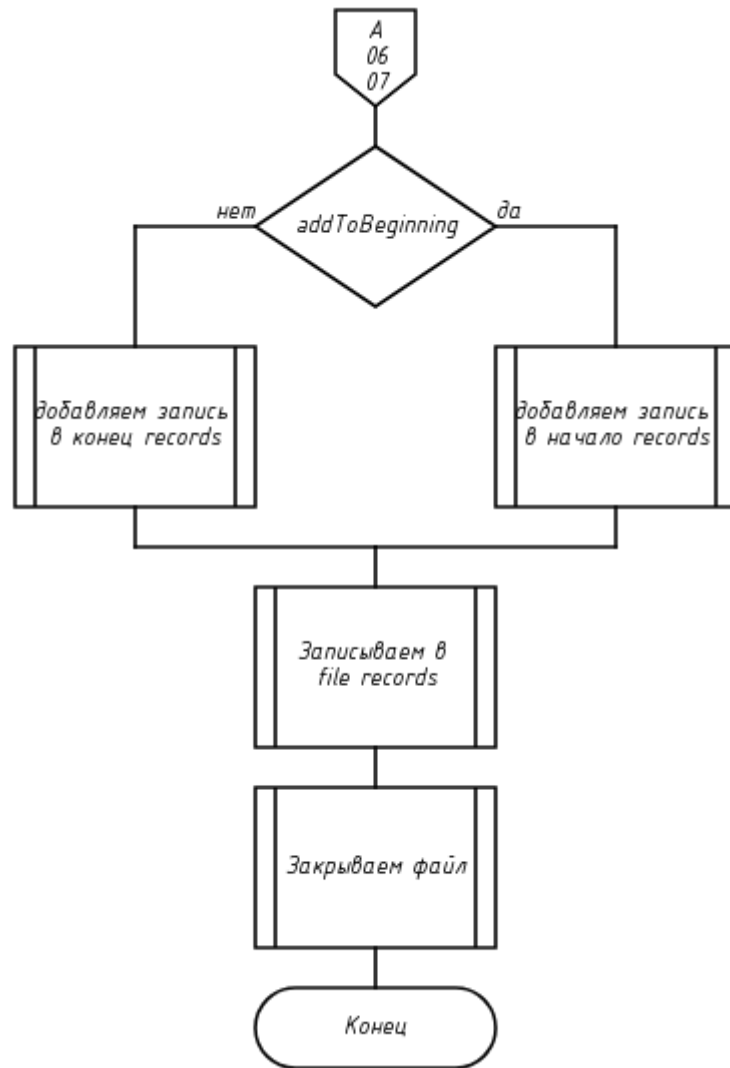


void displayFile(const char *filename, int numRecords)



void addRecord(const char *filename, int &numRecords, bool addToBeginning)



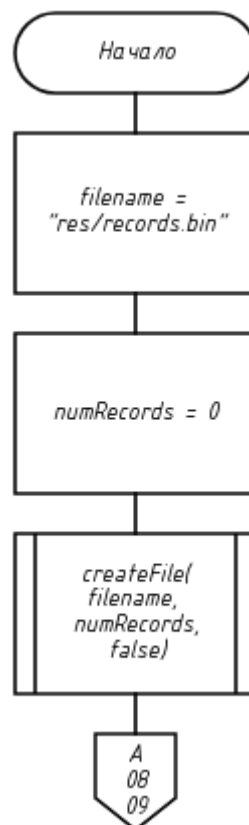


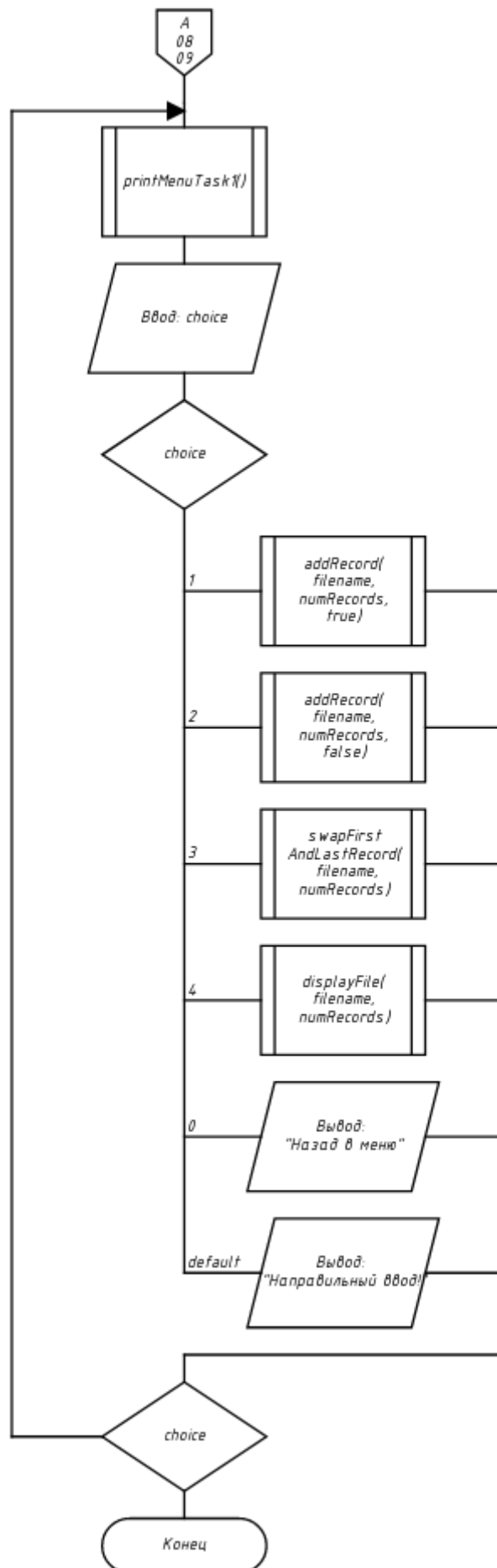
Структура Record

void printMenuTask1()

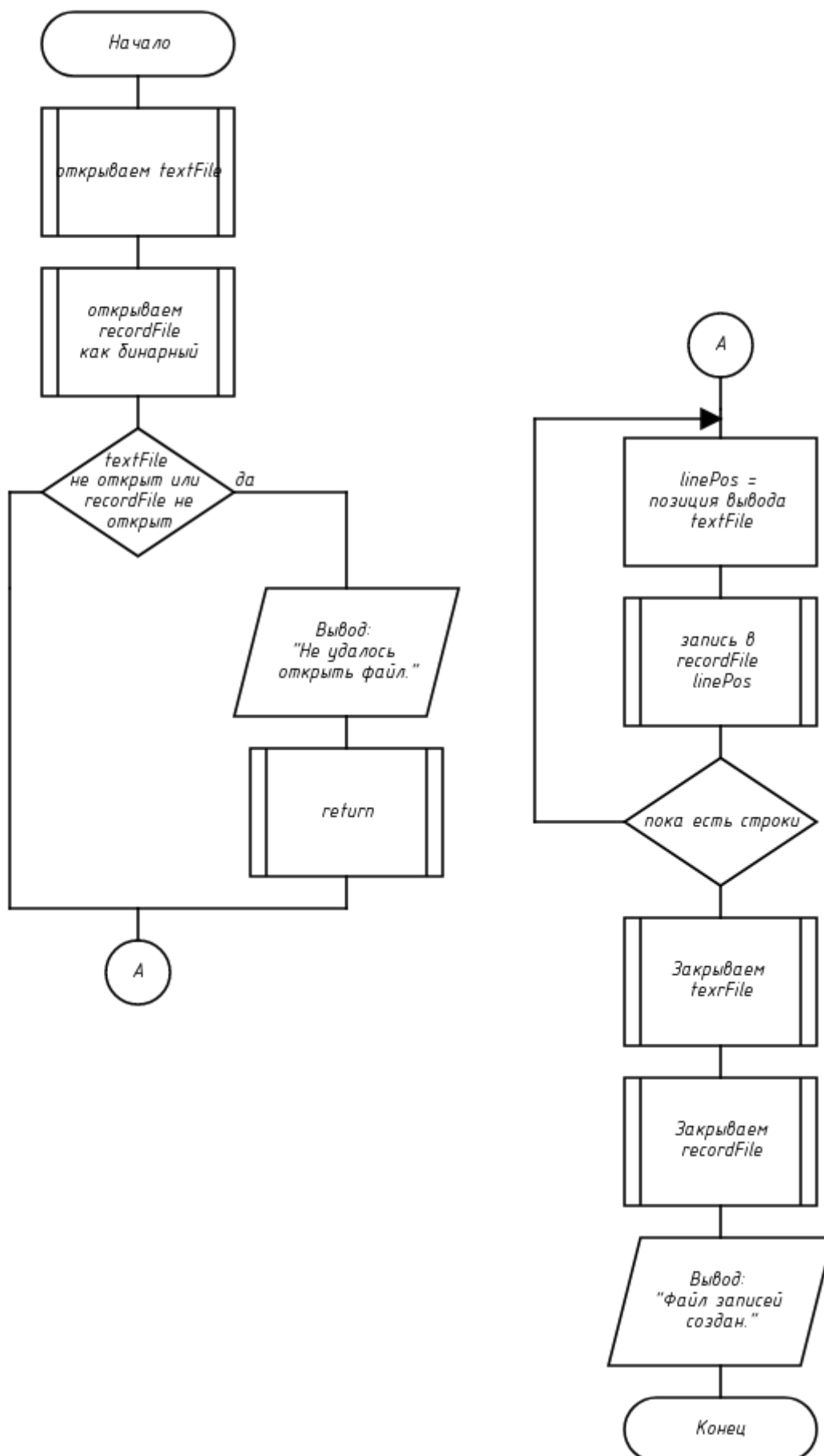


void task_1()

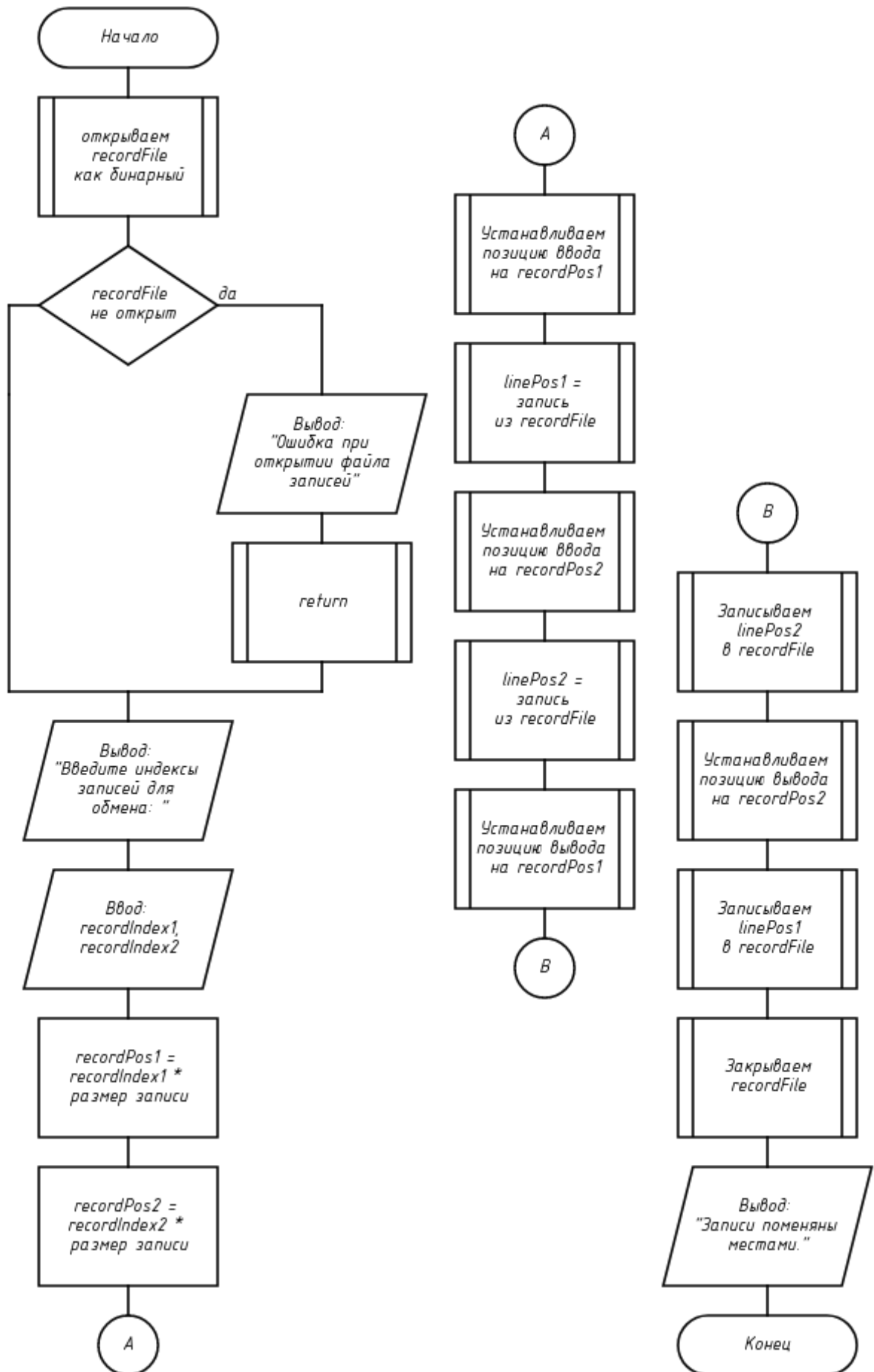




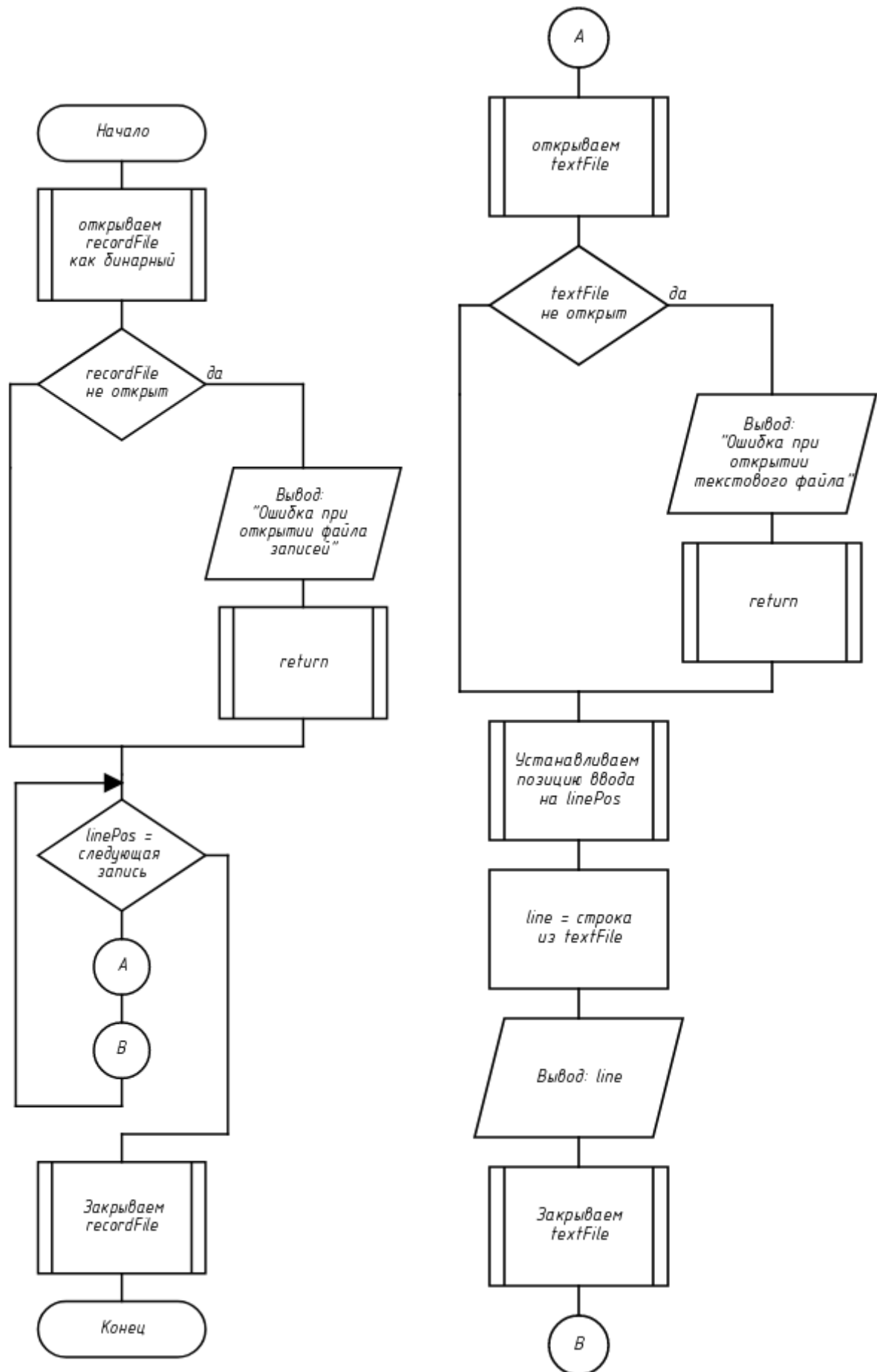
void createRecordFile(const char *textFilePath, const char *recordFilePath)



void swapRecords(const char *recordFilePath)



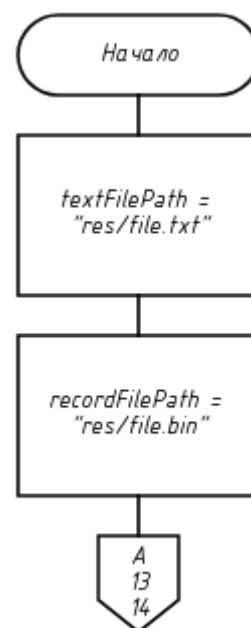
void printRecordFile(const char *recordFilePath)

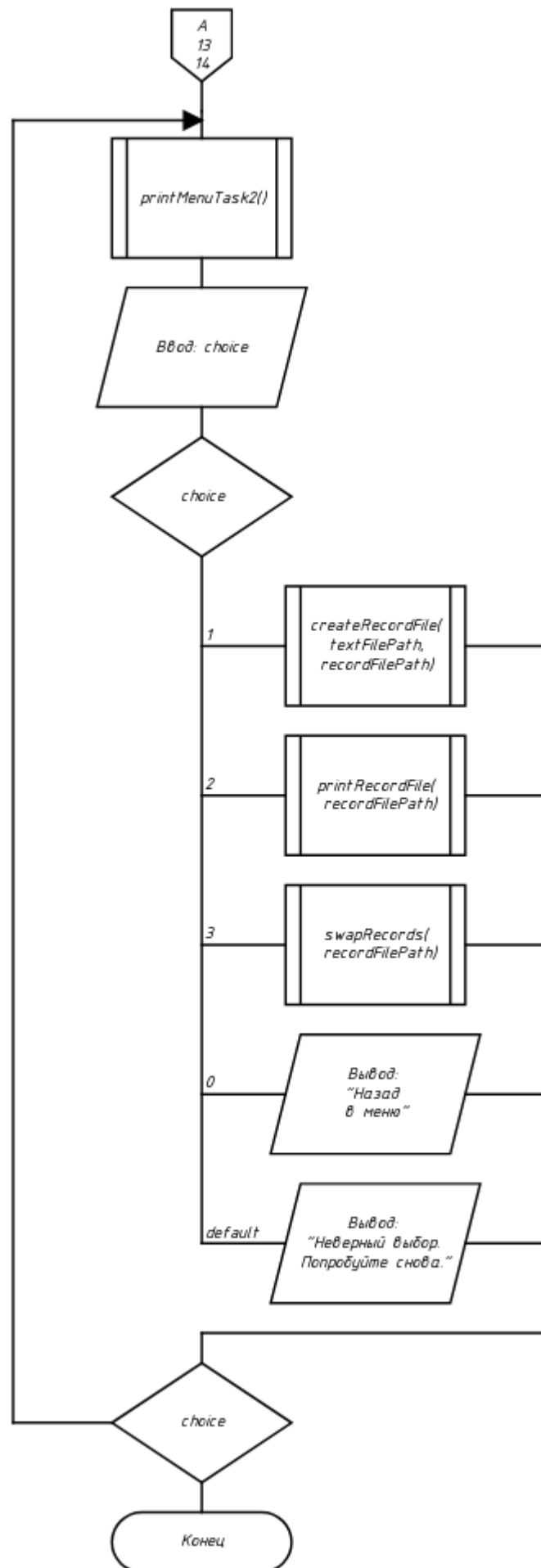


`void printMenuTask2()`

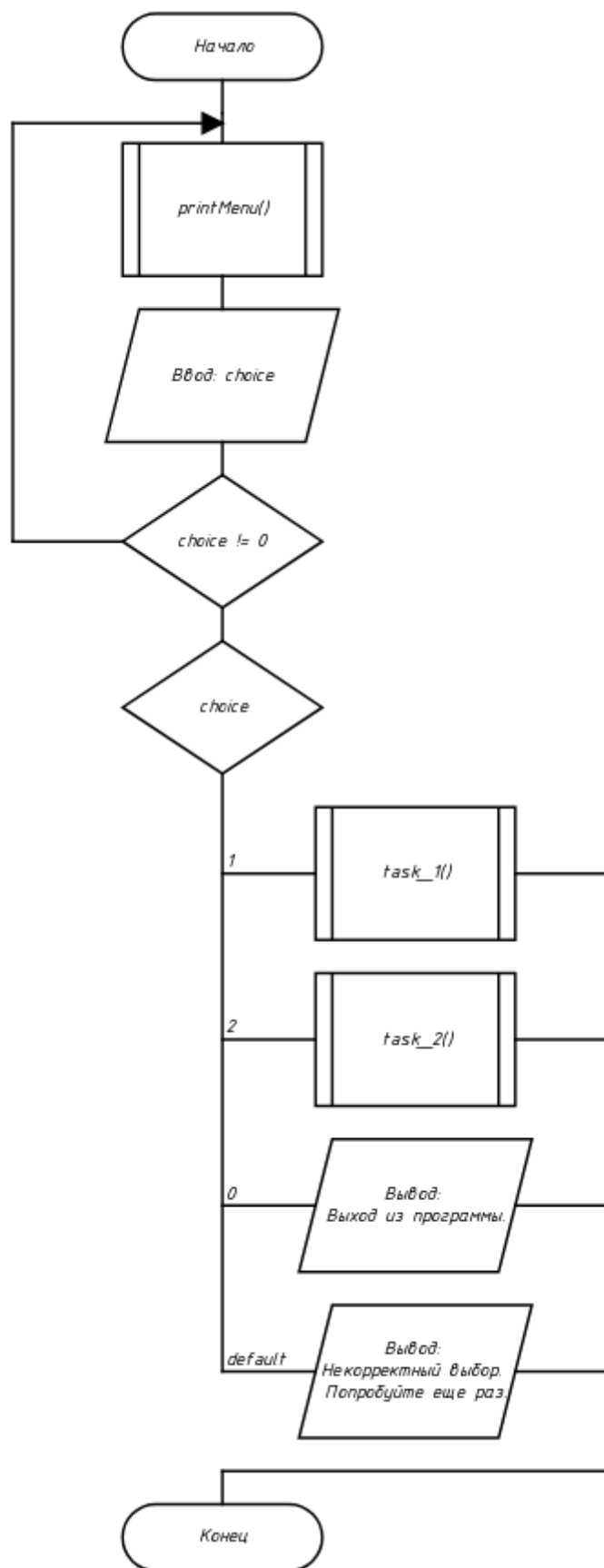


`void task_2()`

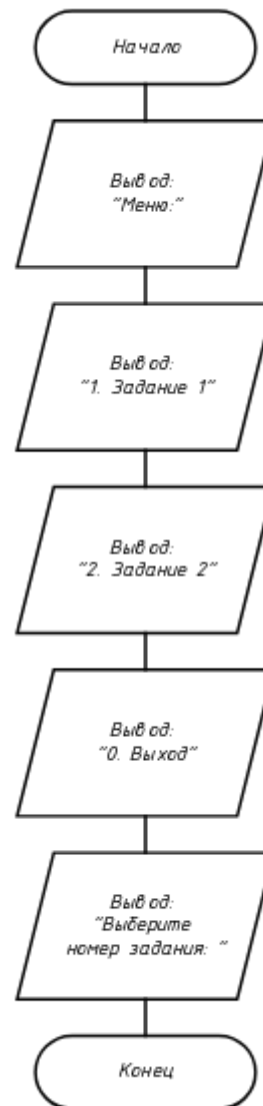




Блок-схема основной программы:



`void printMenu();`



Листинг программы:

Файл task_1.h:

```
1  #ifndef TASK_1_H
2  #define TASK_1_H
3
4  struct Record
5  {
6      int readerTicketNumber;
7      char author[50];
8      char title[50];
9      char orderDate[20];
10 };
11
12 void inputRecord(Record &record);
13 void createFile(const char *filename, int &numRecords, bool byUser);
14 void displayFile(const char *filename, int numRecords);
15 void addRecord(const char *filename, int &numRecords, bool
addToBeginning);
16 void swapFirstAndLastRecord(const char *filename, int numRecords);
17 void printMenuTask1();
18 void task_1();
19
20 #endif // TASK_1_H
```

Файл task_1.cpp:

```
1  #include <iostream>
2  #include <fstream>
3  #include <cstring>
4
5  #include "task_1.h"
6
7  using namespace std;
8
9  void task_1()
10 {
11     const char *filename = "res/records.bin";
12     int numRecords = 0;
13
14     createFile(filename, numRecords, false);
15
16     int choice;
17     do
18     {
19         printMenuTask1();
20         cin >> choice;
21
22         switch (choice)
23         {
24             case 1:
25                 addRecord(filename, numRecords, true);
26                 break;
27             case 2:
28                 addRecord(filename, numRecords, false);
29                 break;
30             case 3:
31                 swapFirstAndLastRecord(filename, numRecords);
32                 break;
33             case 4:
```



```

34         displayFile(filename, numRecords);
35         break;
36     case 0:
37         cout << "Назад в меню" << endl;
38         break;
39     default:
40         cout << "Неправильный ввод!" << endl;
41         break;
42     }
43     cout << endl;
44 } while (choice);
45 }
46
47 void inputRecord(Record &record)
48 {
49     cout << "Введите номер читательского билета: ";
50     cin >> record.readerTicketNumber;
51
52     cout << "Введите автора: ";
53     cin.ignore();
54     cin.getline(record.author, 50);
55
56     cout << "Введите название: ";
57     cin.getline(record.title, 50);
58
59     cout << "Введите дату заказа: ";
60     cin.getline(record.orderDate, 20);
61 }
62
63 void createFile(const char *filename, int &numRecords, bool byUser)
64 {
65     ofstream file(filename, ios::binary);
66     if (!file.is_open())
67     {
68         cout << "Ошибка при создании файла:" << filename << endl;
69         return;
70     }
71
72     numRecords = 10;
73     Record *records = new Record[numRecords];
74
75     if (byUser)
76     {
77         for (int i = 0; i < numRecords; i++)
78         {
79             cout << "Введите запись №" << i + 1 << ":" << endl;
80             Record *record = new Record;
81             inputRecord(*record);
82             records[i] = *record;
83             delete record;
84             cout << endl;
85         }
86     }
87     else
88     {
89         for (int i = 0; i < numRecords; i++)
90         {
91             records[i] = {i + 1, "Author", "Title", "2023-01-01"};
92         }
93     }
94
95     file.write(reinterpret_cast<char *>(records), sizeof(Record) *
numRecords);
96     file.close();

```

```

97     delete[] records;
98 }
99
100 void displayFile(const char *filename, int numRecords)
101 {
102     ifstream file(filename, ios::binary);
103     if (!file.is_open())
104     {
105         cout << "Ошибка при отображении файла:" << filename << endl;
106         return;
107     }
108
109     Record *records = new Record[numRecords];
110     file.read(reinterpret_cast<char *>(records), sizeof(Record) *
numRecords);
111
112     for (int i = 0; i < numRecords; i++)
113     {
114         cout << "Запись № " << i + 1 << ":" << endl;
115         cout << "Номер чит.билета: " << records[i].readerTicketNumber <<
endl;
116         cout << "Автор: " << records[i].author << endl;
117         cout << "Название: " << records[i].title << endl;
118         cout << "Дата заказа: " << records[i].orderDate << endl;
119         cout << endl;
120     }
121
122     file.close();
123
124     delete[] records;
125 }
126
127 void addRecord(const char *filename, int &numRecords, bool
addToBeginning)
128 {
129     fstream file(filename, ios::binary | ios::in | ios::out);
130     if (!file.is_open())
131     {
132         cout << "Ошибка при открытии файла:" << filename << endl;
133         return;
134     }
135
136     Record record;
137     inputRecord(record);
138
139     Record *records = new Record[++numRecords];
140     file.read(reinterpret_cast<char *>(records), sizeof(Record) *
(numRecords - 1));
141
142     if (addToBeginning)
143     {
144         memmove(&records[1], &records[0], sizeof(Record) * (numRecords -
1));
145         memcpy(&records[0], &record, sizeof(Record));
146     }
147     else
148     {
149         memcpy(&records[numRecords - 1], &record, sizeof(Record));
150     }
151
152     file.seekp(0);
153     file.write(reinterpret_cast<char *>(records), sizeof(Record) *
numRecords);

```

```

155         file.close();
156     }
157     delete[] records;
158 }
159
160 void swapFirstAndLastRecord(const char *filename, int numRecords)
161 {
162     fstream file(filename, ios::binary | ios::in | ios::out);
163     if (!file.is_open())
164     {
165         cout << "Ошибка при открытии файла:" << filename << endl;
166         return;
167     }
168
169     Record *records = new Record[numRecords];
170     file.read(reinterpret_cast<char *>(records), sizeof(Record) *
numRecords);
171
172     Record temp = records[0];
173     records[0] = records[numRecords - 1];
174     records[numRecords - 1] = temp;
175
176     file.seekp(0);
177     file.write(reinterpret_cast<char *>(records), sizeof(Record) *
numRecords);
178
179     file.close();
180
181     delete[] records;
182 }
183
184 void printMenuTask1()
185 {
186     cout << "Меню Задания 1:" << endl;
187     cout << "1. Добавить запись в начало" << endl;
188     cout << "2. Добавить запись в конец" << endl;
189     cout << "3. Поменять местами первую и последнюю запись" << endl;
190     cout << "4. Показать записи" << endl;
191     cout << "0. Назад в меню" << endl;
192     cout << "Выбор: ";
193 }
194

```

Файл task_2.h:

```

1  #ifndef TASK_2_H
2  #define TASK_2_H
3
4  const int MAX_LINE_LENGTH = 256;
5  void createRecordFile(const char *textFilePath, const char
*recordFilePath);
6  void swapRecords(const char *recordFilePath);
7  void printRecordFile(const char *recordFilePath);
8  void printMenuTask2();
9  void task_2();
10
11 #endif // TASK_2_H

```

Файл task_2.cpp:

```
1  #include <iostream>
2  #include <fstream>
3  #include <cstring>
4
5  #include "task_2.h"
6
7  using namespace std;
8
9  void task_2()
10 {
11     const char *textFilePath = "res/file.txt";
12     const char *recordFilePath = "res/file.bin";
13
14     int choice;
15     do
16     {
17         printMenuTask2();
18         cin >> choice;
19         switch (choice)
20         {
21             case 1:
22                 createRecordFile(textFilePath, recordFilePath);
23                 break;
24             case 2:
25                 printRecordFile(recordFilePath);
26                 break;
27             case 3:
28                 swapRecords(recordFilePath);
29                 break;
30             case 0:
31                 cout << "Назад в меню" << endl;
32                 break;
33             default:
34                 cout << "Неверный выбор. Попробуйте снова." << endl;
35                 break;
36         }
37         cout << endl;
38     } while (choice != 0);
39 }
40
41 void createRecordFile(const char *textFilePath, const char
*recordFilePath)
42 {
43     ifstream textFile(textFilePath);
44     ofstream recordFile(recordFilePath, ios::binary);
45
46     if (!textFile.is_open() || !recordFile.is_open())
47     {
48         cout << "Не удалось открыть файл." << endl;
49         return;
50     }
51
52     char line[MAX_LINE_LENGTH];
53
54     do
55     {
56         streampos linePos = textFile.tellg();
57         recordFile.write(reinterpret_cast<const char *>(&linePos),
sizeof(streampos));
58     } while (textFile.getline(line, MAX_LINE_LENGTH));
59 }
```

```

60     textFile.close();
61     recordFile.close();
62
63     cout << "Файл записей создан." << endl;
64 }
65
66 void swapRecords(const char *recordFilePath)
67 {
68     fstream recordFile(recordFilePath, ios::binary | ios::in |
ios::out);
69
70     if (!recordFile.is_open())
71     {
72         cout << "Не удалось открыть файл записей." << endl;
73         return;
74     }
75
76     int recordIndex1, recordIndex2;
77     cout << "Введите индексы записей для обмена: ";
78     cin >> recordIndex1 >> recordIndex2;
79
80     streampos recordPos1 = recordIndex1 * sizeof(streampos);
81     streampos recordPos2 = recordIndex2 * sizeof(streampos);
82
83     recordFile.seekg(recordPos1);
84     streampos linePos1;
85     recordFile.read(reinterpret_cast<char *>(&linePos1),
sizeof(streampos));
86
87     recordFile.seekg(recordPos2);
88     streampos linePos2;
89     recordFile.read(reinterpret_cast<char *>(&linePos2),
sizeof(streampos));
90
91     recordFile.seekp(recordPos1);
92     recordFile.write(reinterpret_cast<const char *>(&linePos2),
sizeof(streampos));
93
94     recordFile.seekp(recordPos2);
95     recordFile.write(reinterpret_cast<const char *>(&linePos1),
sizeof(streampos));
96
97     recordFile.close();
98
99     cout << "Записи поменяны местами." << endl;
100 }
101
102 void printRecordFile(const char *recordFilePath)
103 {
104     ifstream recordFile(recordFilePath, ios::binary);
105
106     if (!recordFile.is_open())
107     {
108         cout << "Не удалось открыть файл записей." << endl;
109         return;
110     }
111
112     streampos linePos;
113     while (recordFile.read(reinterpret_cast<char *>(&linePos),
sizeof(streampos)))
114     {
115         ifstream textFile("res/file.txt");
116         if (!textFile.is_open())
117         {

```

```

118         cout << "Не удалось открыть текстовый файл." << endl;
119         return;
120     }
121
122     textFile.seekg(linePos);
123     char line[MAX_LINE_LENGTH];
124     if (textFile.getLine(line, MAX_LINE_LENGTH))
125     {
126         cout << line << endl;
127     }
128
129     textFile.close();
130 }
131
132 recordFile.close();
133 }
134
135 void printMenuTask2()
136 {
137     cout << "Меню:" << endl;
138     cout << "1. Создать файл записей" << endl;
139     cout << "2. Вывести файл записей" << endl;
140     cout << "3. Обменять записи местами" << endl;
141     cout << "0. Назад в меню" << endl;
142     cout << "Введите выбор: ";
143 }

```

Файл main.cpp:

```

1  #include <iostream>
2  #include <fstream>
3  #include <cstring>
4
5  #include "task_1.h"
6  #include "task_2.h"
7
8  using namespace std;
9
10 void printMenu();
11
12 int main()
13 {
14     int choice;
15
16     do
17     {
18         printMenu();
19         cin >> choice;
20
21         switch (choice)
22         {
23             case 1:
24                 task_1();
25                 break;
26             case 2:
27                 task_2();
28                 break;
29             case 0:
30                 cout << "Выход из программы." << endl;
31                 break;
32             default:
33                 cout << "Некорректный выбор. Попробуйте еще раз." << endl;
34                 break;

```

```

35         }
36
37         cout << endl;
38
39     } while (choice != 0);
40 }
41
42 void printMenu()
43 {
44     cout << "Меню:" << endl;
45     cout << "1. Задание 1" << endl;
46     cout << "2. Задание 2" << endl;
47     cout << "0. Выход" << std::endl;
48     cout << "Выберите номер задания: ";
49 }
50

```

Результаты работы:

```

Меню:
1. Задание 1
2. Задание 2
0. Выход
Выберите номер задания: 1
Меню Задания 1:
1. Добавить запись в начало
2. Добавить запись в конец
3. Поменять местами первую и последнюю запись
4. Показать записи
0. Назад в меню
Выбор: 

```

```

Меню Задания 1:
1. Добавить запись в начало
2. Добавить запись в конец
3. Поменять местами первую и последнюю запись
4. Показать записи
0. Назад в меню
Выбор: 1
Введите номер читательского билета: 1
Введите автора: 1
Введите название: 1
Введите дату заказа: 1

Меню Задания 1:
1. Добавить запись в начало
2. Добавить запись в конец
3. Поменять местами первую и последнюю запись
4. Показать записи
0. Назад в меню
Выбор: 4
Запись № 1:
Номер чит.билета: 1
Автор: 1
Название: 1
Дата заказа: 1

```


1. Создать файл записей
2. Вывести файл записей
3. Обменять записи местами
0. Назад в меню

[illegible]

25

Литература

1. Курс лекций доцента кафедры ФН1-КФ Пчелинцевой Н.И.
2. Зырянов, К. И. Программирование на С++: учебное пособие / К. И. Зырянов, Н. П. Кисленко. — Новосибирск: Новосибирский государственный архитектурно-строительный университет (Сибстрин), ЭБС АСВ, 2017. — 129 с. — ISBN 978-5-7795-0817- 9. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://www.iprbookshop.ru/85873.html>
3. Фридман, А. Л. Язык программирования С++: учебное пособие / А. Л. Фридман. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 217 с. — ISBN 978-5-4497-0920-2. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://www.iprbookshop.ru/102076.html>