МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Факультет автоматики и вычислительной техники

Кафедра автоматики



**РАСЧЁТНО-ГРАФИЧЕСКАЯ РАБОТА**

по дисциплине: «Информатика»

вариант № 10

Выполнил: Проверил:

Студент гр. АВТ-219 Ядрышников О.Д

Баскаков М.Д.

«\_\_\_» \_\_\_\_\_\_\_\_\_ 2022 г.«\_\_\_» \_\_\_\_\_\_\_\_\_ 2022 г.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оценка, подпись)

Новосибирск

2022

**Исходные данные и задание для варианта:**

Написать программу, выполняющую действия с базой данных (далее БД). База данных должна быть организована в виде **массива структур**. Память под массив структур выделяется либо **статически**, либо **динамически**. Количество записей ограничено. Программа должна обеспечивать следующий набор операций (**выбор действия осуществляется через меню**):

1. Ввод записи с произвольным номером

2. Вывод записи с заданным номером

3. Сортировка записей по заданному полю в порядке убывания или возрастания

4. Вывод всех записей в отсортированном порядке на экран или принтер

5. Сохранение всех записей в файле

6. Чтение записей из файла

7. Создание двух и более подходящих для БД запросов\* (дополнительное задание для получения дополнительных баллов)

Для **сортировки БД использовать массив указателей**. Ввод и вывод данных должен **сопровождаться необходимыми подсказками**.

**Варианты полей базы данных:**

**A – Фамилия, год рождения, оклад**

**B – Наименование детали, количество, вес**

**C – Название пункта, расстояние, количество рейсов**

**D – Название книги, количество страниц, цена**

**E – Фамилия, номер курса, стипендия**

**F – Название фирмы, количество работников, уставной капитал**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Поля записи БД (элементы структуры) | Кол-во записей в БД | № поля  для сортировки | Выделение памяти под структуры | Тип сортировки | Устройство для вывода записей |
| 10 | E | 100 | 1 | динамически | Возраст | Экран |

**Текст программы:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct {

char lastName[32];

int course, scholarship;

} data;

#define LENGTH 100

data \*db;

data \*\*dbIndex;

int recordsCount;

void printRecord(data \*record);

void sort();

data readRecord();

void createRecord(int index);

int tryReadIndex(int \*index);

void printIndex();

void saveToFile();

void loadFromFile();

void printActions();

void readLine(char \*str, int len);

void main() {

db = malloc(sizeof(data) \* LENGTH);

dbIndex = malloc(sizeof(data \*) \* LENGTH);

recordsCount = 0;

int index;

while (1) {

printActions();

int action;

scanf("%d", &action);

switch (action) {

case 0:

free(db);

free(dbIndex);

return;

case 1:

createRecord(recordsCount);

recordsCount++;

break;

case 2:

if (tryReadIndex(&index))

printRecord(db + index);

else

printf("Record at index %d doesn't exist\n", index);

break;

case 3:

sort();

break;

case 4:

sort();

printIndex();

break;

case 5:

saveToFile();

break;

case 6:

loadFromFile();

break;

case 7:

if (tryReadIndex(&index)) {

printRecord(db + index);

createRecord(index);

} else

printf("Record at index %d doesn't exist\n", index);

break;

case 8:

printf("Now DB contains %d records; capacity - %d\n", recordsCount, LENGTH);

break;

default:

printf("Undefined action\n");

break;

}

}

}

void printRecord(data \*record) {

printf("\tLast name: %s\n", record->lastName);

printf("\tCourse: %d\n", record->course);

printf("\tScholarship: %d\n", record->scholarship);

}

void sort() {

for (int i = 0; i < recordsCount; i++) {

dbIndex[i] = db + i;

}

char wasChanged = 1;

while (wasChanged) {

wasChanged = 0;

for (int i = 0; i < recordsCount - 1; i++)

if (strcmp(dbIndex[i]->lastName, dbIndex[i + 1]->lastName) > 1) {

data \*temp = dbIndex[i];

dbIndex[i] = dbIndex[i + 1];

dbIndex[i + 1] = temp;

wasChanged = 1;

}

}

}

data readRecord() {

data record;

printf("Enter student last name:\n");

getchar();

readLine(record.lastName, 32);

printf("Enter student course:\n");

scanf("%d", &record.course);

printf("Enter student scholarship:\n");

scanf("%d", &record.scholarship);

return record;

}

void createRecord(int index) {

data record = readRecord();

db[index] = record;

printf("Record inserted at index %d\n", index);

}

int tryReadIndex(int \*index) {

printf("Enter index of record\n");

scanf("%d", index);

return \*index >= 0 && \*index < recordsCount;

}

void printIndex() {

for (int i = 0; i < recordsCount; i++) {

printRecord(\*(dbIndex + i));

printf("\n");

}

}

void saveToFile() {

printf("Enter file name to save:\n");

char fileName[32];

scanf("%s", fileName);

FILE \*file = fopen(fileName, "wb+");

if (!file) {

printf("Specified file \"%s\" can't be opened\n", fileName);

return;

}

fwrite(db, sizeof(data), recordsCount, file);

fclose(file);

printf("DB successfully saved to file \"%s\"\n", fileName);

}

void loadFromFile() {

printf("Enter file name to load:\n");

char fileName[32];

scanf("%s", fileName);

FILE \*file = fopen(fileName, "rb");

if (!file) {

printf("Specified file \"%s\" can't be opened\n", fileName);

return;

}

recordsCount = fread(db, sizeof(data), LENGTH, file);

fclose(file);

printf("%d records successfully loaded from file \"%s\"\n", recordsCount, fileName);

}

void printActions() {

printf("Select action from listed below:\n");

printf("0. Exit\n");

printf("1. Create record\n");

printf("2. Print record\n");

printf("3. Sort\n");

printf("4. Print all sorted records\n");

printf("5. Save to file\n");

printf("6. Read from file\n");

printf("7. Edit record\n");

printf("8. Records count\n");

}

void readLine(char \*str, int len) {

int input, index = 0;

while ((input = getchar()) != '\n' && len - 1 > index) {

if (input >= 0)

str[index++] = input;

}

str[index] = 0;

}

**Результаты:**



