

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«Московский Авиационный Институт»

(Национальный Исследовательский Университет)

Факультет: №8 «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная Математика и программирование»

Курсовая работа

По курсу «вычислительные системы»

II семестр

Группа:	М8О-107Б-19
Студент:	Носов Эмиль
Преподаватель:	Сластушенский Юрий Викторович
Оценка:	
Дата:	

## Задание 6: Обработка последовательной файловой структуры

### Цель работы

Разработать последовательную структуру данных для представления простейшей базы данных на файлах в СП Си в соответствии с заданным вариантом. Составить программу генерации внешнего нетекстового файла заданной структуры, содержащего представительный набор записей. Распечатать содержимое сгенерированного файла в виде таблицы и выполнить над ним заданное действие.

### Задание

Вариант 16: Найти фамилии лучших студенток курса (не имеющих отметок ниже четырёх и по сумме баллов не уступающих другим студентам своей группы).

### Алгоритм работы программы

#### Создание, удаление и заполнение базы:

Имя изменяемого файла запрашивается у пользователя. Для создания используется функция `foren`, для удаления — функция `remove`. При добавлении новой записи в файл сначала производится подсчёт номера этой записи, затем добавление записи, запрошенной у пользователя. При удалении записи производится запрос критериев для удаления, после чего запоминаются номера всех записей, соответствующих этим критериям. После этого исходная база данных копируется во временный файл, причем для каждой удаляемой записи запрашивается разрешение на удаление. Если разрешение получено, то запись не копируется во временный файл. После того, как в се нужные значения были сохранены во временном файле, они перезаписываются в основной файл и временный файл удаляется.

#### Печать базы, выполнение запросов и варианта:

Имя обрабатываемого файла запрашивается у пользователя. При печати базы из файла читаются записи и выводятся на экран до тех пор, пока не будет достигнут конец файла. При выполнении запроса пользователь вводит критерии запроса, после чего составляется список номеров записей, удовлетворяющих условиям. После этого необходимые записи выводятся на экран. При выполнении варианта сначала выполняется вычисление максимальной суммы баллов для каждой группы. Затем, пробегаясь по базе данных, мы проверяем, является ли студент девушкой и, если является, то не меньше ли сумма её баллов максимальной сумме в её группе. Если данная запись соответствует требованиям, то фамилия студентки выводится на экран.

### Функции и структуры

В программе используются структуры `record`, хранящая запись базы данных, и `lst`, хранящая массив целых чисел и используемая для того, чтобы хранить список

выбранных студентов.

void printrec(record rec)	Вывод на экран структуры record
void help(int i)	Вывод на экран сообщения help
char * readname()	Чтение строки до символа переноса строки
void create()	Создание файла
void print()	Печать файла
void add()	Добавление записи в файл
void delall()	Удаление файла
int * increase(int * a, int k)	Увеличение размера массива целых чисел, для которого ранее была выделена память
int findgroup(int * groups, int group, int k)	Поиск в массиве целых чисел длиной k числа, стоящего на позиции group
void var16()	Нахождение лучших студенток курса
lst request(char *name)	Поиск студентов, удовлетворяющих некоторым условиям
void printlist(int * list, int l)	Вывод массива, содержащего номера выбранных студентов
void req()	Выполнение запроса к базе данных и вывод его результатов на экран
void del()	Удаление записи из базы данных

## Код программы

### кп6.h

```
#ifndef DB_H
#define DB_H
#include <stdio.h>
#include <stdlib.h>

typedef struct record record;
typedef struct lst lst;

struct record{
    int number;
    char lname[80];
    char inert[6];
    char sex;
    int group;
    int grades[6];
};

int eq(char* a, char* b){
    int i=0,j=0,l=1;
```

```

        while(a[i]!='\0')i++;
        while(b[j]!='\0')j++;
        if (i==j){
            for(int k=0; k<i; k++) if(a[k]!=b[k]) l=0;
        }else l=0;
        return(l);
    }

int equal(record a, record b){
    int i=0;
    if(eq(a.lname, b.lname)&&eq(a.inert, b.inert)\
    &&(a.sex==b.sex)&&(a.group==b.group))i=1;
    for(int k=0; k<6; k++) if(a.grades[k]!=b.grades[k]) i=0;
    return(i);
}

void printrec(record rec){
    printf("%d %s %s %c %d"\
    ,rec.number, rec.lname, rec.inert, rec.sex, rec.group);
    for(int i=0; i<6;i++){
        printf(" %d", rec.grades[i]);
    }
    printf("\n");
}

void help(int i){
    if(i==1) printf("This program can print database, request records from it and find
best female students.\n");
    else printf("This program can create and delete databases and records in it.\n");
}

char * readname(){
    int i=0;
    char b, *a=(char*)malloc(sizeof(char)*256);
    scanf("%c",&b);
    if(b=='\n'){
        scanf("%c",&b);
    }
    while((b!='\n')&&(i!=256)){
        a[i]=b;
        i++;
        scanf("%c",&b);
    }
    a[i]='\0';
    return(a);
}

void create() {
    record rec;
    printf("Enter name of database:");
    char *name=readname();
    int i=0;
    FILE *file=fopen(name,"wb");
    if(file!=NULL)printf("Database was created");
    else printf("Database wasn't created");
    if (fclose(file)<0) printf("Wasn't closed");
    free(name);
}

void print() {
    int k;
    record rec;

```

```

        printf("Enter name of database:");
        char *name = readname();
        int i;
        FILE * file = fopen(name,"rb");
        if(file!=NULL){
            while(fread(&rec,sizeof(record),1,file)) printrec(rec);
            k=fclose(file);
            free(name);
        }
        else printf("Database wasn't created");
    }

void add() {
    record rec;
    printf("Enter name of database:");
    char *name=readname();
    int k=0;
    FILE *file=fopen(name,"rb");
    if(file!=NULL){while(fread(&rec,sizeof(record),1,file)) k=rec.number;
    fclose(file);
    rec.number=k+1;
    file=fopen(name,"ab");
    printf("Record should have this structure:\n\
    [last name] [initials] [sex] [number of cgroup] [grade 1] ... [grade 6]\n\
    Enter record:");
    scanf("%s %s %c %d",rec.lname,rec.inert, &rec.sex, &rec.group);
    rec.inert[5]='\0';
    for(int i=0; i<6;i++){
        scanf("%d",&rec.grades[i]);
    }
    printrec(rec);
    fwrite(&rec,sizeof(record),1,file);
    if(fclose(file)<0) printf("Wasn't closed");
    free(name);
    }
    else printf("This database wasn't created.");
}

void delall() {
    record rec;
    printf("Enter name of database:");
    char *name=readname();
    int i=0;
    remove(name);
    FILE *f=fopen(name,"rb");
    if(f==NULL) printf("Data base was deleted");
    else printf("Database wasn't deleted");
    free(name);
}

int * increase(int * a, int k){
    int *b=(int*)malloc((k+10)*sizeof(int));
    for(int i; i<k; i++){
        b[i]=a[i];
    }
    return(b);
}

int findgroup(int * groups, int group, int k){
    int i=0;
    while((groups[i]!=group)&&(i<k)) i++;
    if(groups[i]!=group)i=-1;
}

```

```

        return(i);
    }

void var16() {
    record rec;
    printf("Enter name of database:");
    char *name = readname();
    int i=-1,k=10,g;
    int *groups, *grades;
    groups=(int*)malloc(sizeof(int)*10);
    grades=(int*)malloc(sizeof(int)*10);
    FILE *file=fopen(name,"rb");
    if(file!=NULL){
        while(fread(&rec,sizeof(record),1,file)){
            g=findgroup(groups, rec.group, k);
            if(g<0) {
                i++;
                groups[i]=rec.group;
                grades[i]=0;
                g=i;
                if(i+1==k){
                    groups=increase(groups, k);
                    grades=increase(grades, k);
                    k+=10;
                }
            }
            for(int j=1; j<6;j++) rec.grades[0]+=rec.grades[j];
            if(rec.grades[0]>grades[g]) grades[g]=rec.grades[0];
        }
        fseek(file, 0, SEEK_SET);
        while(fread(&rec,sizeof(record),1,file)){
            i=0;
            g=findgroup(groups, rec.group, k);
            for(int j=1; j<6; j++){
                rec.grades[0]+=rec.grades[j];
                if (rec.grades[j]<4) i=1;
            }
            if((i==0)&&(rec.grades[0]>=grades[g])&&\
                ((rec.sex=='F')||(rec.sex=='f')))\
                printf("%s\n", rec.lname);
        }
        fclose(file);
        free(name);
    }
    else printf("This file wasn't created");
}

struct lst{
    int *list;
    int l;
    int j;
};

lst request(char *name) {
    record rec;
    lst L;
    L.list=(int*)malloc(10*sizeof(int));
    L.l=9;
    L.j=0;
    int i=0, n=0, v=0;
    char p;

```

```

FILE *file=fopen(name,"rb");
if(file!=NULL){
    printf("Parameters:\ng - Grades\n\
l - lname\ni - initials\nn - number\n\
s - Sex\nG - Group\nEnter type of parameter:");
    scanf("%c",&p);
    switch(p){
        case 'l':
            printf("Enter student's name:");
            name=readname();
            i=1;
            break;
        case 'g':
            i=2;
            printf("Enter subject's number:");
            scanf("%d",&n);
            printf("Enter value:");
            scanf("%d",&v);
            break;
        case 'i':
            i=3;
            printf("Enter student's initials:");
            name=readname();
            break;
        case 'n':
            i=4;
            printf("Enter student's number:");
            scanf("%d",&n);
            break;
        case 's':
            i=5;
            printf("Enter student's sex (F or M):");
            scanf("\n%c",&p);
            break;
        case 'G':
            i=6;
            printf("Enter group's number:");
            scanf("%d",&n);
            break;
    }
    while(fread(&rec,sizeof(record),1,file)){
        if(L.l==L.j){
            L.list=increase(L.list, L.l);
            L.l+=10;
        }
        switch(i){
            case 1:
                if(eq(name, rec.lname)){
                    L.list[L.j]=rec.number;
                    L.j++;
                }
                break;
            case 2:
                if(rec.grades[n-1]==v){
                    L.list[L.j]=rec.number;
                    L.j++;
                }
                break;
            case 3:
                if(eq(name, rec.inert)){
                    L.list[L.j]=rec.number;
                    L.j++;
                }
            }
        }
    }
}

```

```

        }
        break;
    case 4:
        if(n==rec.number){
            L.list[L.j]=rec.number;
            L.j++;
        }
        break;
    case 5:
        if(p==rec.sex){
            L.list[L.j]=rec.number;
            L.j++;
        }
        break;
    case 6:
        if(n==rec.group){
            L.list[L.j]=rec.number;
            L.j++;
        }
        break;
    }
    }
    fclose(file);
}
else printf("Database wasn't created");
return L;
}

void printlist(int * list, int l){
    for(int i=0; i<l; i++){
        printf("%d ",list[i]);
    }
    printf("\n");
}

void req(){
    int i=0;
    record rec;
    printf("Enter name of database:");
    char* name=readname();
    FILE *file=fopen(name,"rb");
    if(file!=NULL){
        fclose(file);
        list L=request(name);
        file=fopen(name,"rb");
        while(fread(&rec,sizeof(record),1,file)){
            if((rec.number==L.list[i])&&(i<=L.j)){
                i++;
                printrec(rec);
            }
        }
        free(name);
        free(L.list);
        fclose(file);
    }
    else printf("File wasn't created");
}

void del() {
    record rec;
    rec.number=0;

```



```

printf("Enter name of database:");
char *name = readname();
int i=0, j=0;
char q;
lst L=request(name);
FILE *file=fopen(name,"rb");
if(file!=NULL){
    FILE *tmp=tmpfile();
    while(fread(&rec, sizeof(record),1,file)){
        if(rec.number==L.list[j]){
            printrec(rec);
            printf("Do you want to delete this record? (Y or
N)");

            scanf("\n%c", &q);
            printf("%c",q);
            if(q=='Y') {
                printf("continue");
                continue;
            }
            if(j<L.j)j++;
        }
        i++;
        rec.number=i;
        fwrite(&rec, sizeof(record),1,tmp);
    }
    fclose(file);
    rewind(tmp);
    file=fopen(name,"wb");
    while(fread(&rec, sizeof(record),1,tmp)){
        fwrite(&rec,sizeof(record),1,file);
    }
    fclose(file);
    fclose(tmp);
    free(name);
}
else printf("Database wasn't created");
}

#endif

```

## creator.c

```

#include "κπ6.h"

int main(){
char a = ' ';
printf("h - Help\na - Add\nd - Delete\nr - Remove base\nc - Create base\n\nEnter q to quit\n\n");
while( a != 'q' ){
if(a != '\n') printf("Enter command:");
scanf("%c",&a);
switch(a){
    case 'h':
        help();
        break;
    case 'a':
        add();
        break;
    case 'd':
        del();
        break;
    case 'r':
        delall();

```

```

        break;
    case 'c':
        create();
        break;
    case 'q':
        return 0;
}
printf("\nOperation %c done!", a);
}
}

```

## request.c

```

#include "kn6.h"

int main(){
    char a = ' ';
    printf("h - Help\np - Print\nr - Request\no - execute option 16\n\nEnter q to quit\n\n");
    while( a != 'q' ){
        if(a != '\n') printf("Enter command:");
        scanf("%c",&a);
        switch(a){
            case 'h':
                help();
                break;
            case 'p':
                print();
                break;
            case 'r':
                req();
                break;
            case 'o':
                var16();
                break;
            case 'q':
                return 0;
        }
        if(a!='\n')printf("\nOperation %c done!\n\n", a);
    }
}

```

## Вывод программы

**\$ ./creator**

h - Help

a - Add

d - Delete

r - Remove base

c - Create base

Enter q to quit

Enter command:c

Enter name of database:2course

Database was created

Operation c done!

Enter command:q

**\$ ls**

2course creator.c kp6.h request

creator request.c sec

**\$ ./creator**

h - Help

a - Add

d - Delete

r - Remove base

c - Create base

Enter q to quit

Enter command:r

Enter name of database:2course

Data base was deleted

Operation r done!

Enter command:q

**\$ ls**

creator request.c sec

creator.c kp6.h request

**\$ ./request**

h - Help

p - Print

r - Request

o - Find

Enter q to quit

Enter command:p

Enter name of database:sec

1 Nosov E.L. M 7 3 3 4 3 3 4

2 Nosova E.L. F 7 5 5 5 5 5 5

3 Holopova E.D. F 6 5 5 5 5 5 5

4 Glykova J.D. F 5 5 5 5 5 5 5

Operation p done!

Enter command:r

Enter name of database:sec

Parameters:

g - Grades

l - lname

i - initials  
n - number  
s - Sex  
G - Group  
Enter type of parameter:s  
Enter student's sex (F or M):M  
1 Nosov E.L. M 7 3 3 4 3 3 4

Operation r done!

Enter command:o  
Enter name of database:sec  
Nosova  
Holopova  
Glykova

Operation o done!

Enter command:q  
**\$ ./creator**  
h - Help  
a - Add  
d - Delete  
r - Remove base  
c - Create base

Enter q to quit

Enter command:a  
Enter name of database:sec  
Record should have this structure:  
[last name] [initials] [sex] [number of cgroup] [grade 1] ... [grade 6]  
Enter record:Krot A.S. F 5 6 5 5 5 5  
5 Krot A.S. F 5 6 5 5 5 5

Operation a done!

Enter name of database:sec  
Record should have this structure:  
[last name] [initials] [sex] [number of cgroup] [grade 1] ... [grade 6]  
Enter record:Trekhova G.P. F 6 4 5 5 5 5 6  
5 Trekhova G.P. F 6 4 5 5 5 5 6

Operation a done!

Enter command:a  
Enter name of database:notexist  
This database wasn't created.  
Operation a done!

Enter command:d  
Enter name of database:sec  
Parameters:  
g - Grades  
l - lname  
i - initials  
n - number  
s - Sex  
G - Group  
Enter type of parameter:l  
Enter student's name:Nosov  
1 Nosov E.L. M 7 3 3 4 3 3 4  
Do you want to delete this record? (Y or N )Y

Operation d done!

Enter command:q  
**\$ ./request**  
h - Help  
p - Print  
r - Request  
o - Find

Enter q to quit

Enter command:o  
Enter name of database:sec  
Nosova  
Holopova  
Krot  
Trekhova

Operation o done!

Enter command:p  
Enter name of database:sec  
1 Nosova E.L. F 7 5 5 5 5 5 5  
2 Holopova E.D. F 6 5 5 5 5 5 5  
3 Glykova J.D. F 5 5 5 5 5 5 5  
4 Krot A.S. F 5 6 5 5 5 5 5

5 Trekhova G.P. F 6 4 5 5 5 6

Operation p done!

Enter command:q

### **Заключение**

В ходе данной работы была создана последовательная структура данных и созданы программы для работы с базами данных на бинарных файлах, использующие эту структуру в качестве записей. Были реализованы создание и удаление базы данных, добавление и удаление записей, произведение запроса, печать базы данных и нахождение лучших студенток курса.