**Правительство Российской Федерации**

**Федеральное государственное автономное образовательное учреждение высшего профессионального образования   
"Национальный исследовательский университет   
"Высшая школа экономики"**

Московский институт электроники и математики Национального

исследовательского университета "Высшая школа экономики"

Департамент прикладной математики

**ОТЧЕТ**

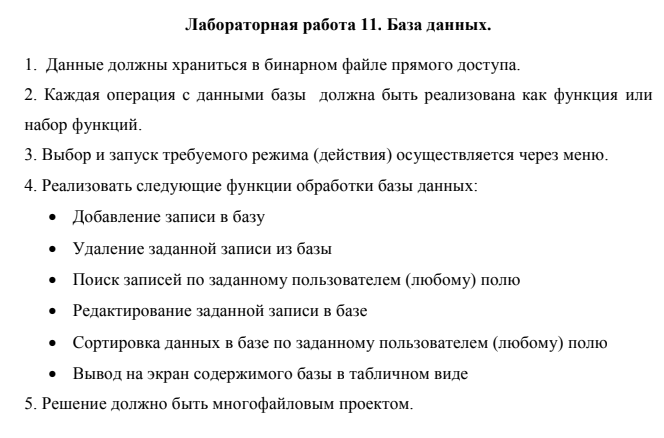
**По лабораторной работе № 11**

**По курсу «Алгоритмизация и программирование»**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  | ФИО студента | | Номер группы | Дата |
| Индюченко Никита Андреевич | БПМ211 | 27.04.2022 |
|  |
|  |
|  |

**Москва – 2022 г.**

**ЗАДАНИЕ (вариант № 12)**

****

*Текст задания Вашего варианта*

**

**  
РЕШЕНИЕ**

Код программы с комментариями

Main.c

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <math.h>

#include "Football\_league.h"

Club database[100];

int index\_club = 0;

// --------- Функции удаления, переноса и вывода клуба ----

int delete\_data(); // удалить запись

void move\_database\_to\_file();//перенос данных из массива в файл

void print\_data(int index); // вывод заданного клуба

int print\_all\_data(); // вывод всех данных

// ---------- Фунции нахождения по полю -------------------

int search\_data();// поиск из всей базы

void search\_name\_trainer(); // поиск по имени тренера

void search\_beetween\_goals\_on\_of(); // поиск по разнице мячей

void search\_score(); // поиск по очкам

void search\_name\_city(); // поиск по названию города

void search\_name\_club(); // поиск по названию клуба

// ----------- Функции добавления поля --------------------

void add\_name\_trainer(); // добавить имя тренера

void add\_name\_Club(); // добавить название клуба

void add\_name\_city(); // добавить название города

int add\_data(); // добавить запись

// ----------- Функции для переименовки поля --------------

int rename\_data();

void rename\_name\_trainer(int\* index); // переименовать тренера

void rename\_beetween\_goals\_on\_of(int\* index); // изменить разницу мячей

void rename\_score(int\* index); // изменить кол-во очков

void rename\_city(int\* index); // переименовать город

void rename\_club(int\* index); // переименовать клуб

// ----------- Функции сортировки -------------------------

int sort\_data(); // сортировка по любому полю. Порядок расположение клубов меняется в исходном массиве !!!

int main()

{

while(1)

{

printf("List commands : \n1 - Adding information\n2 - Deleting information\n3 - Element search\n");

printf("4 - Editing element\n5 - Sorting\n6 - Print all database\n7 - Console cleaning\n0 - Exit\n");

int command;

scanf\_s("%d", &command);

switch (command)

{

case 1:

add\_data();

// добавление записи в базу - Сделано!

break;

case 2:

delete\_data();

// удаление заданной записи из базы - Сделано!

break;

case 3:

search\_data();

// поиск записей по заданному пользователю (любому) полю - Сделано!

break;

case 4:

rename\_data();

// редактирование заданной записи в базе (любое поле) - Сделано!

break;

case 5:

sort\_data();

// сортировка данных в базе по заданному пользователем (любому) полю - Сделано !

break;

case 6:

print\_all\_data();

// вывод на экран содержимого базы в табличном виде

break;

case 0:

// выход

exit(1);

break;

case 7:

system("cls"); // очистка окна

break;

default:

printf("This command isn't working\n");

break;

}

}

}

int search\_data()

{

if (index\_club == 0)

{

printf("No data!\n");

return 1;

}

printf("Select point\n");

int choice = 0;

while (1)

{

printf("1 - name\_club\n2 - name\_city\n3 - score\n4 - beetween\_goals\_on\_of\n5 - name\_trainer\nYour choice : ");

scanf\_s("%d", &choice);

switch (choice)

{

case 1:

// поиск клуба - Cделано!

search\_name\_club();

return 0;

case 2:

search\_name\_city();

// поиск города - Сделано!

return 0;

case 3:

search\_score();

// поиск по очкам - Сделано!

return 0;

case 4:

search\_beetween\_goals\_on\_of();

// поиск по разницам забитых мячей - Сделано!

return 0;

case 5:

search\_name\_trainer();

// поиск по имени тренера - Сделано!

return 0;

default:

printf("This command doesn't exist\n");

break;

}

}

}

void search\_name\_trainer()

{

char\* tmp\_name\_trainer = (char\*)malloc(sizeof(char) \* 256);

char tmp\_char;

int int\_switch = 0, tr = 0, tmp\_index = 0;

printf("Name trainer : ");

scanf\_s("%c", &tmp\_char);

for (int i = 0; i < 256; i++)

{

scanf\_s("%c", &tmp\_char);

if (tmp\_char == '\n')

{

tmp\_index = i;

tmp\_char = 0;

tmp\_name\_trainer[i] = tmp\_char;

break;

}

tmp\_name\_trainer[i] = tmp\_char;

}

for (int i = 0; i < index\_club; i++)

{

tr = 0;

for (int l = 0; l <= tmp\_index; l++)

{

if (database[i].name\_trainer[l] != tmp\_name\_trainer[l])

{

tr = 1;

break;

}

}

if (tr == 0)

{

int\_switch = 1;

print\_data(i);

}

}

free(tmp\_name\_trainer);

if (int\_switch == 1) printf("Successfully found!\n");

else printf("There is no such data\n");

}

void search\_beetween\_goals\_on\_of()

{

printf("beetween goals on of : ");

int tmp\_goals, int\_switch = 0;

scanf\_s("%d", &tmp\_goals);

for (int i = 0; i < index\_club; i++)

{

if (database[i].beetween\_goals\_on\_of == tmp\_goals)

{

int\_switch = 1;

print\_data(i);

}

}

if (int\_switch == 1) printf("Successfully found!\n");

else printf("There is no such data\n");

}

void search\_score()

{

printf("Score : ");

int tmp\_score, int\_switch = 0;

scanf\_s("%d", &tmp\_score);

for (int i = 0; i < index\_club; i++)

{

if (database[i].score == tmp\_score)

{

int\_switch = 1;

print\_data(i);

}

}

if (int\_switch == 1) printf("Successfully found!\n");

else printf("There is no such data\n");

}

void search\_name\_city()

{

char\* tmp\_name\_city = (char\*)malloc(sizeof(char) \* 256);

char tmp\_char;

int tmp\_index, int\_switch = 0;

printf("Name city : ");

scanf\_s("%c", &tmp\_char);

for (int i = 0; i < 256; i++)

{

scanf\_s("%c", &tmp\_char);

if (tmp\_char == '\n')

{

tmp\_index = i;

tmp\_char = 0;

tmp\_name\_city[i] = tmp\_char;

break;

}

tmp\_name\_city[i] = tmp\_char;

}

int tr = 0;

for (int i = 0; i < index\_club; i++)

{

tr = 0;

for (int l = 0; l <= tmp\_index; l++)

{

if (database[i].name\_city[l] != tmp\_name\_city[l])

{

tr = 1;

break;

}

}

if (tr == 0)

{

int\_switch = 1;

print\_data(i);

}

}

free(tmp\_name\_city);

if (int\_switch == 1) printf("Successfully found!\n");

else printf("There is no such data\n");

}

void search\_name\_club()

{

char\* tmp\_name\_club = (char\*)malloc(sizeof(char) \* 256);

char tmp\_char;

int tmp\_index, int\_switch = 0;

printf("Name club : ");

scanf\_s("%c", &tmp\_char);

for (int i = 0; i < 256; i++)

{

scanf\_s("%c", &tmp\_char);

if (tmp\_char == '\n')

{

tmp\_index = i;

tmp\_char = 0;

tmp\_name\_club[i] = tmp\_char;

break;

}

tmp\_name\_club[i] = tmp\_char;

}

int tr = 0; // есть ли такая строка

for (int i = 0; i < index\_club; i++)

{

tr = 0;

for (int l = 0; l <= tmp\_index; l++)

{

if (database[i].name\_club[l] != tmp\_name\_club[l])

{

//printf("%c", database[i].name\_club[l]);

tr = 1; // если не подходит

break;

}

}

if (tr == 0)

{

int\_switch = 1;

// вывод подходящей записи

print\_data(i);

}

}

free(tmp\_name\_club);

if (int\_switch == 1) printf("Successfully found!\n");

else printf("There is no such data\n");

}

void print\_data(int index)

{

printf("name Club : ");

for (int j = 0; j < 256; j++)

{

if (database[index].name\_club[j] == 0) break;

printf("%c", database[index].name\_club[j]);

}

printf("\nname city : ");

for (int j = 0; j < 256; j++)

{

if (database[index].name\_city[j] == 0) break;

printf("%c", database[index].name\_city[j]);

}

printf("\nscore : %d", database[index].score);

printf("\nbeetween goals on of : %d", database[index].beetween\_goals\_on\_of);

printf("\nname traner : ");

for (int j = 0; j < 256; j++)

{

if (database[index].name\_trainer[j] == 0) break;

printf("%c", database[index].name\_trainer[j]);

}

printf("\n");

}

int delete\_data()

{

if (index\_club == 0)

{

printf("No data\n");

return 2;

}

int index\_deleted\_data;

printf("Enter number data :");

scanf\_s("%d", &index\_deleted\_data);

if (index\_deleted\_data >= index\_club)

{

printf("This data isn't in base\n");

return 1;

}

for (int i = index\_deleted\_data; i < index\_club - 1; i++)

{

for (int j = 0; j < 256; j++)

{

\*(database[i].name\_club + j) = \*(database[i + 1].name\_club + j);

}

for (int j = 0; j < 256; j++)

{

\*(database[i].name\_city + j) = \*(database[i + 1].name\_city + j);

}

database[i].score = database[i + 1].score;

database[i].beetween\_goals\_on\_of = database[i + 1].beetween\_goals\_on\_of;

for (int j = 0; j < 256; j++)

{

\*(database[i].name\_trainer + j) = \*(database[i + 1].name\_trainer + j);

}

}

index\_club--;

printf("Database changed\n");

move\_database\_to\_file();

}

void add\_name\_trainer()

{

char tmp;

char\* tmp\_name\_trainer = (char\*)malloc(sizeof(char) \* 256);

scanf\_s("%c", &tmp);

int end = 0;

for (int i = 0; i < 256; i++)

{

scanf\_s("%c", &tmp);

if (tmp == '\n')

{

tmp = 0;

tmp\_name\_trainer[i] = tmp;

break;

}

tmp\_name\_trainer[i] = tmp;

end++;

}

for (int i = 0; i <= end; i++)

{

\*(database[index\_club].name\_trainer + i) = tmp\_name\_trainer[i];

}

free(tmp\_name\_trainer);

}

void add\_name\_Club()

{

char tmp;

char\* tmp\_name\_Cl = (char\*)malloc(sizeof(char) \* 256);

scanf\_s("%c", &tmp);

int end = 0;

for (int i = 0; i < 256; i++)

{

scanf\_s("%c", &tmp);

if (tmp == '\n')

{

tmp = 0;

tmp\_name\_Cl[i] = tmp;

break;

}

tmp\_name\_Cl[i] = tmp;

end++;

}

for (int i = 0; i <= end; i++)

{

\*(database[index\_club].name\_club + i) = tmp\_name\_Cl[i];

}

free(tmp\_name\_Cl);

}

void add\_name\_city()

{

char tmp;

char\* tmp\_name\_city = (char\*)malloc(sizeof(char) \* 256);

int end = 0;

for (int i = 0; i < 256; i++)

{

scanf\_s("%c", &tmp);

if (tmp == '\n')

{

tmp = 0;

tmp\_name\_city[i] = tmp;

break;

}

tmp\_name\_city[i] = tmp;

end++;

}

for (int i = 0; i <= end; i++)

{

\*(database[index\_club].name\_city + i) = tmp\_name\_city[i];

}

free(tmp\_name\_city);

}

int add\_data()

{

if (index\_club == 100)

{

printf("User didn't can to create or to add new club, Because there is no memory access\n");

return 1;

}

printf("Fill in the information\n");

printf("Name Club : ");

add\_name\_Club();

//puts(database[index\_club].name\_club);

printf("Name city : ");

add\_name\_city();

printf("Score : ");

scanf\_s("%d", &database[index\_club].score);

printf("Beetween\_goals\_on\_of : ");

scanf\_s("%d", &database[index\_club].beetween\_goals\_on\_of);

database[index\_club].score = abs(database[index\_club].score);

database[index\_club].beetween\_goals\_on\_of = abs(database[index\_club].beetween\_goals\_on\_of);

printf("Coach's name : ");

add\_name\_trainer();

index\_club++;

printf("Data saved!\n");

move\_database\_to\_file();

}

void rename\_club(int\* index)

{

printf("Name club : ");

char\* tmp\_name\_Cl = (char\*)malloc(sizeof(char) \* 256);

char tmp;

scanf\_s("%c", &tmp);

int end = 0;

for (int i = 0; i < 256; i++)

{

scanf\_s("%c", &tmp);

if (tmp == '\n')

{

tmp = 0;

tmp\_name\_Cl[i] = tmp;

break;

}

tmp\_name\_Cl[i] = tmp;

end++;

}

for (int i = 0; i <= end; i++)

{

\*(database[\*index].name\_club + i) = tmp\_name\_Cl[i];

}

free(tmp\_name\_Cl);

printf("Change completed!\n");

move\_database\_to\_file();

}

void rename\_city(int\* index)

{

printf("Name city : ");

char\* tmp\_name\_city = (char\*)malloc(sizeof(char) \* 256);

char tmp;

scanf\_s("%c", &tmp);

int end = 0;

for (int i = 0; i < 256; i++)

{

scanf\_s("%c", &tmp);

if (tmp == '\n')

{

tmp = 0;

tmp\_name\_city[i] = tmp;

break;

}

tmp\_name\_city[i] = tmp;

end++;

}

for (int i = 0; i <= end; i++)

{

\*(database[\*index].name\_city + i) = tmp\_name\_city[i];

}

free(tmp\_name\_city);

printf("Change completed!\n");

move\_database\_to\_file();

}

void rename\_score(int\* index)

{

printf("Score : ");

int tmp\_score = 0;

scanf\_s("%d", &tmp\_score);

database[\*index].score = abs(tmp\_score);

printf("Change completed!\n");

move\_database\_to\_file();

}

void move\_database\_to\_file()

{

FILE\* f = fopen("f.bin", "wb");

for (int i = 0; i < index\_club; i++) // перенос данных из массива в файл

{

fwrite(database[i].name\_club, sizeof(char) \* 256, 1, f);

fwrite(database[i].name\_city, sizeof(char) \* 256, 1, f);

fwrite(&database[i].score, sizeof(int), 1, f);

fwrite(&database[i].beetween\_goals\_on\_of, sizeof(int), 1, f);

fwrite(&database[i].name\_trainer, sizeof(char) \* 256, 1, f);

}

fclose(f);

}

void rename\_beetween\_goals\_on\_of(int\* index)

{

printf("beetween goals on of : ");

int tmp\_goals = 0;

scanf\_s("%d", &tmp\_goals);

database[\*index].beetween\_goals\_on\_of = abs(tmp\_goals);

printf("Change completed!\n");

move\_database\_to\_file();

}

void rename\_name\_trainer(int\* index)

{

printf("Name trainer : ");

char\* tmp\_name\_trainer = (char\*)malloc(sizeof(char) \* 256);

char tmp;

scanf\_s("%c", &tmp);

int end = 0;

for (int i = 0; i < 256; i++)

{

scanf\_s("%c", &tmp);

if (tmp == '\n')

{

tmp = 0;

tmp\_name\_trainer[i] = tmp;

break;

}

tmp\_name\_trainer[i] = tmp;

end++;

}

for (int i = 0; i <= end; i++)

{

\*(database[\*index].name\_trainer + i) = tmp\_name\_trainer[i];

}

free(tmp\_name\_trainer);

printf("Change completed!\n");

move\_database\_to\_file();

}

int rename\_data()

{

if (index\_club == 0)

{

printf("No data!\n");

return 2;

}

printf("Enter the number data : ");

int index\_data;

scanf\_s("%d", &index\_data);

if (index\_data >= index\_club)

{

printf("This data doesn't in database\n");

return 1;

}

int choice;

while (1)

{

printf("1 - name\_club\n2 - name\_city\n3 - score\n4 - beetween\_goals\_on\_of\n5 - name\_trainer\nYour choice : ");

scanf\_s("%d", &choice);

switch (choice)

{

case 1:

rename\_club(&index\_data);

// изменить клуб - Сделано !

return 0;

case 2:

rename\_city(&index\_data);

// изменить город - Сделано !

return 0;

case 3:

rename\_score(&index\_data);

// изменить кол-во очков - Сделано !

return 0;

case 4:

rename\_beetween\_goals\_on\_of(&index\_data);

// изменить разницу забитых мячей

return 0;

case 5:

rename\_name\_trainer(&index\_data);

// изменить имя тренера - Сделано !

return 0;

default:

printf("This command doesn't exist\n");

break;

}

}

}

int sort\_data()

{

if (index\_club == 0)

{

printf("No data!\n");

return 2;

}

printf("Select the field you want to sort by : \n");

int choice = 0;

printf("1 - Sort by clubs\n2 - Sort by city\n3 - Sort by score\n4 - Sort by goals\n5 - Sort by name trainer\n");

printf("Your choice : ");

scanf\_s("%d", &choice);

if ((choice > 5) || (choice < 1))

{

printf("This command isn't working\n");

return 1;

}

Club\* tmp = (Club\*)malloc(sizeof(Club));

for (int L = 0; L < index\_club - 1; L++)

{

for (int i = 0; i < index\_club - 1; i++)

{

if ((((database[i].score > database[i + 1].score) && (choice == 3)) || ((database[i].beetween\_goals\_on\_of > database[i + 1].beetween\_goals\_on\_of) && (choice == 4)) || ((strcmp(database[i].name\_club, database[i + 1].name\_club) > 0) && (choice == 1))) || ((strcmp(database[i].name\_city, database[i + 1].name\_city) > 0) && (choice == 2)) || ((strcmp(database[i].name\_trainer, database[i + 1].name\_trainer) > 0) && (choice == 5)))

{

// перенос данных о клубе во временный клуб

for (int j = 0; j < 256; j++)

{

tmp[0].name\_club[j] = database[i].name\_club[j];

if (database[i].name\_club[j] == 0) break;

}

for (int j = 0; j < 256; j++)

{

tmp[0].name\_city[j] = database[i].name\_city[j];

if (database[i].name\_city[j] == 0) break;

}

tmp[0].score = database[i].score;

tmp[0].beetween\_goals\_on\_of = database[i].beetween\_goals\_on\_of;

for (int j = 0; j < 256; j++)

{

tmp[0].name\_trainer[j] = database[i].name\_trainer[j];

if (database[i].name\_trainer[j] == 0) break;

}

// перенос данных i+1 клуба

for (int j = 0; j < 256; j++)

{

database[i].name\_club[j] = database[i + 1].name\_club[j];

if (database[i].name\_club[j] == 0) break;

}

for (int j = 0; j < 256; j++)

{

database[i].name\_city[j] = database[i + 1].name\_city[j];

if (database[i].name\_city[j] == 0) break;

}

database[i].score = database[i + 1].score;

database[i].beetween\_goals\_on\_of = database[i + 1].beetween\_goals\_on\_of;

for (int j = 0; j < 256; j++)

{

database[i].name\_trainer[j] = database[i + 1].name\_trainer[j];

if (database[i].name\_trainer[j] == 0) break;

}

// перенос данных из tmp в клуб

for (int j = 0; j < 256; j++)

{

database[i + 1].name\_club[j] = tmp[0].name\_club[j];

if (database[i + 1].name\_club[j] == 0) break;

}

for (int j = 0; j < 256; j++)

{

database[i + 1].name\_city[j] = tmp[0].name\_city[j];

if (database[i + 1].name\_city[j] == 0) break;

}

database[i + 1].score = tmp[0].score;

database[i + 1].beetween\_goals\_on\_of = tmp[0].beetween\_goals\_on\_of;

for (int j = 0; j < 256; j++)

{

database[i + 1].name\_trainer[j] = tmp[0].name\_trainer[j];

if (database[i + 1].name\_trainer[j] == 0) break;

}

}

}

}

printf("Sort!\n");

print\_all\_data();

free(tmp);

}

int print\_all\_data()

{

if (index\_club == 0)

{

printf("No Data!\n");

return 1;

}

printf("-------------All Data-----------------\n");

int max\_size\_name\_club = 0, max\_size\_name\_city = 0, max\_size\_name\_trainer = 0;

for (int i = 0; i < index\_club; i++)

{

if (max\_size\_name\_club < strlen(database[i].name\_club)) max\_size\_name\_club = strlen(database[i].name\_club);

if (max\_size\_name\_city < strlen(database[i].name\_city)) max\_size\_name\_city = strlen(database[i].name\_city);

if (max\_size\_name\_trainer < strlen(database[i].name\_trainer)) max\_size\_name\_trainer = strlen(database[i].name\_trainer);

}

printf("N | ");// Номер

printf("Club");

for (int i = 0; i < max\_size\_name\_club; i++) printf(" ");

printf("| ");

printf("City");

for (int i = 0; i < max\_size\_name\_city; i++) printf(" ");

printf("| ");

printf("Score | ");

printf("Beetween\_goals\_on\_of | Trainer");

for (int i = 0; i < max\_size\_name\_trainer; i++) printf(" ");

printf("|\n");

for (int i = 0; i < index\_club; i++)

{

int k = 0;

int tmp\_value = i + 1;

while (tmp\_value > 0)

{

tmp\_value /= 10;

k++;

}

printf("%d", i + 1);

for (int j = 0; j < 8 - k; j++) printf(" ");

printf("| ");

for (int j = 0; j < max\_size\_name\_club + 4; j++)

{

if (j < strlen(database[i].name\_club))

{

printf("%c", database[i].name\_club[j]);

}

else

{

printf(" ");

}

}

printf("| ");

for (int j = 0; j < max\_size\_name\_city + 4; j++)

{

if (j < strlen(database[i].name\_city))

{

printf("%c", database[i].name\_city[j]);

}

else

{

printf(" ");

}

}

printf("| ");

tmp\_value = database[i].score;

k = 0;

while (tmp\_value > 0)

{

tmp\_value /= 10;

k++;

}

printf("%d", database[i].score);

for (int j = 0; j < 12 - k; j++)

{

printf(" ");

}

printf("| ");

k = 0;

tmp\_value = database[i].beetween\_goals\_on\_of;

while (tmp\_value > 0)

{

tmp\_value /= 10;

k++;

}

printf("%d", database[i].beetween\_goals\_on\_of);

for (int j = 0; j < 20 - k; j++)

{

printf(" ");

}

printf(" | ");

for (int j = 0; j < max\_size\_name\_trainer + 7; j++)

{

if (j < strlen(database[i].name\_trainer))

{

printf("%c", database[i].name\_trainer[j]);

}

else

{

printf(" ");

}

}

printf("| ");

printf("\n");

}

}

Football\_league.h

#pragma once

struct Football\_Club

{

char name\_club[256]; // название клубы

char name\_city[256]; // название города

int score; // кол-во набранных очков

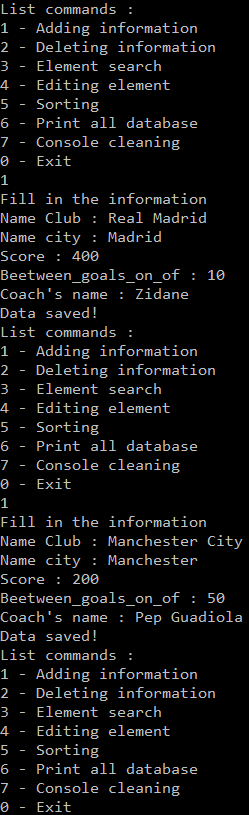
int beetween\_goals\_on\_of;// разница забитых и пропущенных мячей

char name\_trainer[256]; // имя тренера

} typedef Club;

**ТЕСТЫ**

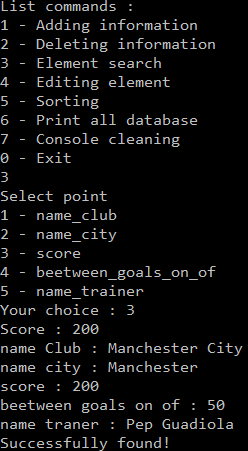
**Будем считать, что работаем с этими данными, чтобы не повторять заполнение.**

****

**Тест № 1**

**Поиск клубов по очкам**

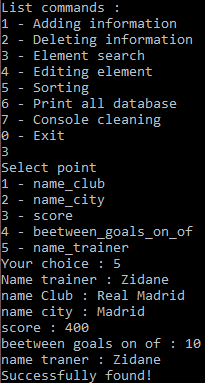
*Результаты теста 1*

**

**Тест № 2**

**Поиск клуба по имени главного тренера**

*Результаты теста 2*

**

**Тест № 3**

**Удалим какой-нибудь клуб (допустим первый) и выведем всю базу**

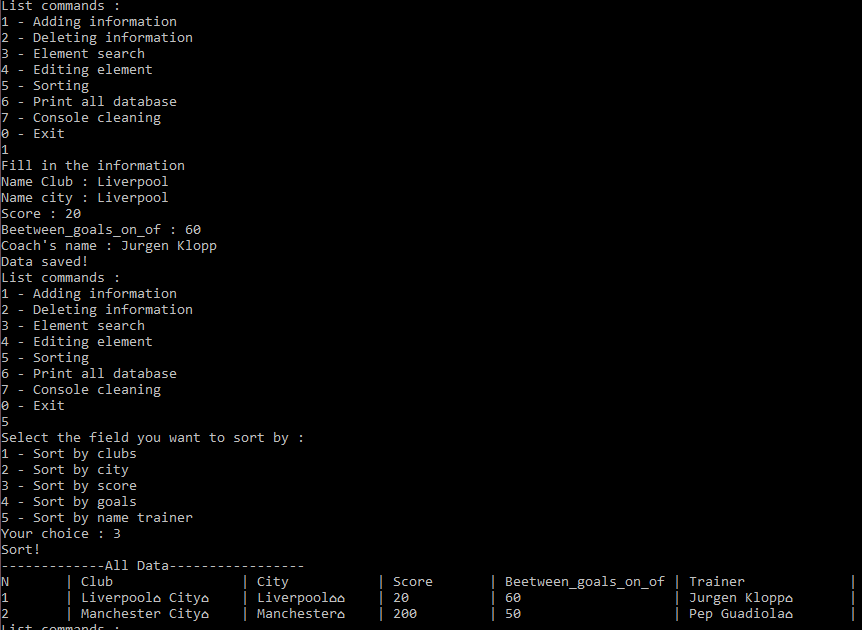
*Результаты теста 3*

****

**Тест № 4**

**Добавим ещё один клуб, также сортируем по очкам**

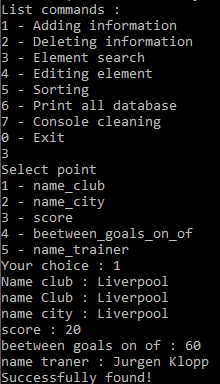
*Результаты теста 4*

****

**Тест № 5**

**Поиск по клубу**

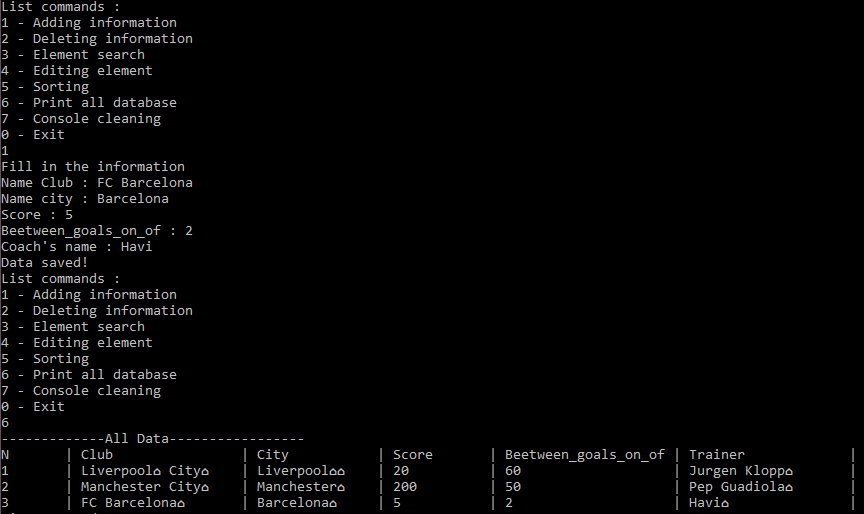
*Результаты теста 5*

****

**Тест №6**

**Добавление клуба, вывод всей базы, которая не сортированная**

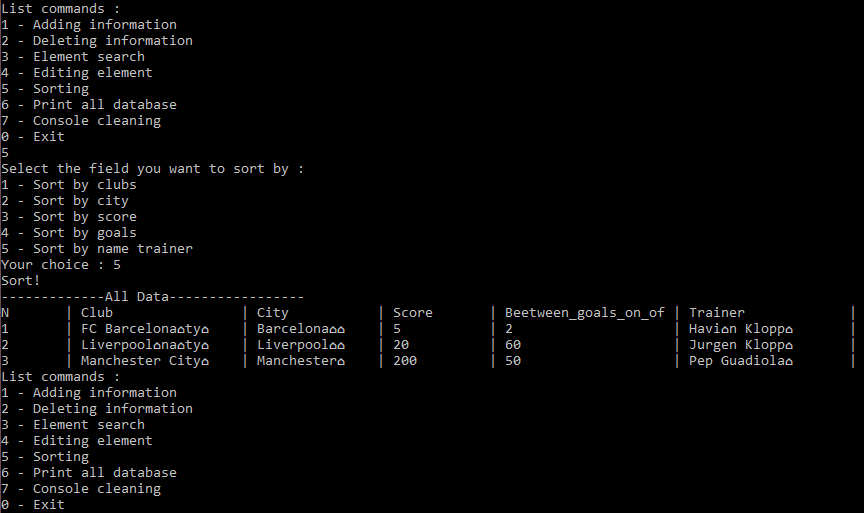
*Результаты теста 6*

****

**Тест №7**

**Сортировка по именам тренеров**

*Результаты теста 7*

****

**Тест № …**

*Результаты теста …*