Балтийский государственный технический университет  
«ВОЕНМЕХ» им. Д. Ф. Устинова

Кафедра О7 «Информационные системы и программная инженерия»

**Практическая работа №1**по дисциплине «Структуры и организация данных»  
на тему «Линейные структуры данных»  
часть 1 «Связный линейный список»  
  
вариант 25

Выполнил:  
Студент Фокин В. К.  
Группа И924Б  
  
Преподаватель:  
Гладевич А. А.

Санкт-Петербург  
2023 г.

Уровень сложности – ***Низкий***.

Данные хранятся в бинарном файле записей, а для обработки считываются в односвязный линейный список (если файл не существует, то создается пустой список). При выходе из программы обработанные данные сохраняются в том же файле. Имя файла с данными задается константой или вводится с клавиатуры. Для взаимодействия с пользователем должно использоваться меню. Обязательные операции для списка: добавление элемента в начало списка, удаление первого элемента списка, просмотр списка, поиск в соответствии с индивидуальным вариантом (две функции). Вывод списка на экран можно выполнять в любом (главное, читабельном) виде. В качестве примера выполнения задания можно использовать программу, хранящуюся в папке с примерами в файле list\_client.c.

Поля данных: название страны, континент, площадь, численность населения, столица. Определить, какие страны на заданном континенте имеют наибольшую и наименьшую плотность населения. Вывести название столицы заданной страны.

Созданные типы данных:

1. DataType – обозначает структуру country

Объявление:

struct country  
{  
 char name[30];   
 char continent[20];   
 float square;  
 float population;  
 char capital[20];  
};

typedef struct country DataType;

Содержит поля структуры country:

name – название страны

continent – название континента

square – площадь страны

population – население страны

capital – столица страны

1. list – обозначает указатель на структуру node

Объявление:

struct node  
{  
 DataType data;  
 struct node \*next;   
};

typedef struct node \*list;

Содержит поля структуры country, а next представляет указатель на следующий объект структуры node.

Используемые функции:

1. Создание и сохранение файла

int write\_file (char \*filename, list begin);

Входные данные – название файла, дескриптор списка

Выходные данные – если запись в файл прошла успешно, возвращается 1, иначе 0

1. Чтение файла

list read\_file (char \*filename);

Входные данные – название файла

Выходные данные – дескриптор списка

1. Заполнение структуры country

DataType input\_place (void);

Входные данные отсутствуют

Выходные данные – заполнение структуры

1. Проверка на ввод букв

void letter\_check (char \*test)

Входные данные – название страны, континента, столицы

Выходные данные отсутствуют

1. Удаление списка

void delete\_list (list begin);

Входные данные – дескриптор списка

Выходные данные отсутствуют

1. Вывод всех данных

void show (list cur);

Входные данные – дескриптор списка

Выходные данные отсутствуют

1. Вывод столицы заданной страны

void show\_capital (list cur);

Входные данные – дескриптор списка

Выходные данные отсутствуют

1. Вывод плотности населения заданного континента

void show\_density (list cur);

Входные данные – дескриптор списка

Выходные данные отсутствуют

1. Добавление узла в начало списка

list new\_node (list begin, DataType country);

Входные данные – дескриптор списка и структура

Выходные данные – дескриптор списка

10. Удаление узла из списка

list delete\_node (list begin);

Входные данные – дескриптор списка

Выходные данные – дескриптор списка

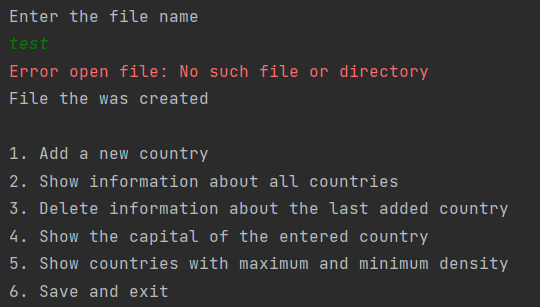
Текст программы:

#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
#include <ctype.h>  
  
struct country //создание структуры страна  
{  
 char name[30]; //название страны  
 char continent[20]; //континент, где страна расположена  
 long square; //площадь страны  
 long population; //население страны  
 char capital[20]; //столица страны  
};  
  
typedef struct country DataType; //создаем синоним к структуре  
  
struct node //создание структуры узел  
{  
 DataType data;  
 struct node \*next; //указатель на следующий элемент  
};  
  
typedef struct node \*list; //указатель на узел  
  
int write\_file (char \*filename, list); //создание и сохранение файла  
list read\_file (char \*filename); //чтение файла  
DataType input\_place (void); //заполнение структуры страна  
void letter\_check (char \*test); //проверка на ввод букв  
void delete\_list (list); //удаление списка  
void show (list); //вывод данных  
void show\_capital (list); //вывод столицы заданной страны  
void show\_density (list); //вывод плотности населения на континенте  
list new\_node (list, DataType); //добавление узла в начало списка  
list delete\_node (list); //удаление узла из списка  
  
int main()  
{  
 char file[30];  
 char menu;  
 list location;  
 puts("Enter the file name");  
 gets(file);  
 location = read\_file(file);  
 do  
 {  
 puts("1. Add a new country");  
 puts("2. Show information about all countries");  
 puts("3. Delete information about the last added country");  
 puts("4. Show the capital of the entered country");  
 puts("5. Show countries with maximum and minimum density");  
 puts("6. Save and exit");  
 menu = getchar();getchar();  
 switch (menu)  
 {  
 case '1': location = new\_node (location, input\_place()); break; //добавление информации о стране  
 case '2': show (location); break; //вывод всего списка  
 case '3': location = delete\_node (location);break; //удаление последнего добавленного узла  
 case '4': show\_capital(location); break; //вывод столицы заданной страны  
 case '5': show\_density(location); break; //вывод максимальной и минимальной плотности стран на заданном континенте  
 }  
 }  
 while (menu!='6');  
 if (write\_file (file, location)) //сохранение файла  
 puts ("File saved");  
 else  
 puts ("File not saved");  
 delete\_list (location); //очистка списка  
 return 0;  
}  
  
int write\_file (char \*filename, list begin)  
{  
 FILE \*f;  
 if ((f = fopen(filename, "wb")) == NULL)  
 {  
 perror("Error create file");  
 getchar();  
 return 0;  
 }  
 while (begin)  
 {  
 if (fwrite (&begin->data, sizeof(DataType), 1, f)) //запись в файл  
 begin = begin->next; //переход к следующему узлу  
 else  
 return 0;  
 }  
 return 1;  
}  
  
list read\_file (char \*filename)  
{  
 FILE \*f;  
 DataType place;  
 list begin = NULL, cur;  
 if ((f = fopen (filename, "rb")) == NULL)  
 {  
 perror ("Error open file");  
 printf("File the was created\n");  
 getchar();  
 return begin;  
 }  
 if (fread(&place, sizeof(place), 1, f))  
 begin = new\_node (NULL, place);  
 else  
 return NULL;  
 cur = begin;  
 while (fread(&place, sizeof(place), 1, f))  
 {  
 cur->next = new\_node (NULL, place);  
 cur = cur->next;  
 }  
 fclose(f);  
 return begin;  
}  
  
DataType input\_place (void)  
{  
 char test[20];  
 long y;  
 DataType place; //структура страна  
 getchar();  
  
 puts("Name of the country"); //ввод названия страны  
 letter\_check(test); //проверка на ввод  
 strcpy(place.name, test); //копирование строки в название страны  
 memset(test, 0, sizeof(test)); //очистка строки  
  
 puts("Continent"); //ввод названия континента  
 letter\_check(test); //проверка на ввод  
 strcpy(place.continent, test); //копирование строки в название континента  
 memset(test, 0, sizeof(test)); //очистка строки  
  
 puts("Square of the country"); //ввод площади страны  
 y = 0;  
 do  
 {  
 scanf("%ld", &y);  
 if (y > 0 && y < 20000000)  
 {  
 place.square = y;  
 }  
 else  
 {  
 puts("Enter a value without letters between 0 and 20000000");  
 y = 0;  
 }  
 }  
 while (!(y > 0 && y < 20000000));  
 fflush(stdin);  
  
 puts("Population"); //ввод населения страны  
 y = 0;  
 do  
 {  
 scanf("%ld", &y);  
 if (y > 0 && y < 2000000000)  
 {  
 place.population = y;  
 }  
 else  
 {  
 puts("Enter a value without letters between 0 and 2000000000");  
 y = 0;  
 }  
 }  
 while (!(y > 0 && y < 2000000000));  
 fflush(stdin);  
  
 puts("Capital of the country"); //ввод столицы страны  
 letter\_check(test); //проверка на ввод  
 strcpy(place.capital, test); //копирование строки в название столицы  
 memset(test, 0, sizeof(test)); //очистка строки  
  
 getchar();  
 return place;  
}  
  
void letter\_check (char \*test)  
{  
 int j, len, cnt = 0;  
 while (!cnt)  
 {  
 gets(test);  
 len = strlen(test);  
 for (j = 0; j < len; j++)  
 {  
 if (!isalpha(test[j]) && !isspace(test[j]))  
 {  
 cnt = 0;  
 break;  
 }  
 cnt = 1; //если символ не буква и не пробел, увеличиваем счетчик  
 }  
 if (!cnt)  
 {  
 printf("The value must not have numbers. Try again.\n");  
 }  
 }  
}  
  
void delete\_list (list begin)  
{  
 list temp = begin; //перенос указателя в начало списка  
 while (temp) //пока в списке есть элементы  
 {  
 begin = temp->next; //перенос указателя на следующий элемент  
 free(temp); //удаление первого элемента  
 temp = begin;  
 }  
}  
  
void show (list cur)  
{  
 int k = 0; //счетчик элементов  
 if (cur == NULL)  
 {  
 puts("List is empty");  
 getchar();  
 return;  
 }  
 puts("| N | Name | Continent | Square | Population | Capital |");  
 puts("----------------------------------------------------------------------------------");  
 while(cur)  
 {  
 printf ("|%2d |%15s |%14s |%13ld |%13ld |%12s |\n", ++k, cur->data.name,  
 cur->data.continent, cur->data.square, cur->data.population, cur->data.capital); //вывод информации  
 cur = cur->next; //перенос указателя на следующий элемент  
 }  
 puts("----------------------------------------------------------------------------------");  
}  
  
void show\_capital (list cur)  
{  
 char ctr[20];  
 char cpt[20];  
 int cnt = 0;  
 if (cur == NULL)  
 {  
 puts("List is empty");  
 getchar();  
 return;  
 }  
 printf("Enter the name of the country\n");  
 scanf("%s", ctr); //ввод названия страны, столицу которой нужно вывести  
 while (cur)  
 {  
 if (!strcmp(cur->data.name, ctr)) //условие вывода столицы  
 {  
 cnt += 1; //если строки соответствуют, увеличиваем счетчик  
 strcpy(cpt, cur->data.capital); //и копируем в cpt название столицы  
 }  
 cur = cur->next; //перенос указателя  
 }  
 if (cnt > 0) //если счетчик больше 0, выводим  
 {  
 puts ("| Name | Capital |");  
 puts ("--------------------------------");  
 printf ("|%15s |%12s |\n", ctr, cpt);  
 puts ("--------------------------------");  
 }  
 else //иначе такой страны нет в списке  
 {  
 printf("There is no such country in the list\n");  
 }  
 getchar();  
}  
  
void show\_density (list cur)  
{  
 char cntnt[20];  
 long mind = 100000;  
 long maxd = 0;  
 char max[50];  
 char min[50];  
 int cnt = 0;  
 if (cur == NULL)  
 {  
 puts("List is empty");  
 getchar();  
 return;  
 }  
 printf("Enter the continent\n");  
 scanf("%s", cntnt); //ввод названия континента, в котором будем искать плотность  
 while (cur)  
 {  
 if (!strcmp(cur->data.continent, cntnt))  
 {  
 cnt += 1; //если строки соответствуют, увеличиваем счетчик  
 if (cur->data.population / cur->data.square > maxd) //если плотность стран выше максимальной имеющейся  
 {  
 maxd = cur->data.population / cur->data.square; //приравниваем ее к наибольшей плотности  
 strcpy(max, cur->data.name); //и копируем в max название страны  
 }  
 if (cur->data.population / cur->data.square < mind) //если плотность стран ниже минимальной имеющейся  
 {  
 mind = cur->data.population / cur->data.square; //приравниваем ее к наименьшей  
 strcpy(min, cur->data.name); //и копируем в min название страны  
 }  
 }  
 cur = cur->next; //перенос указателя  
 }  
 if (cnt > 0) //если счетчик больше нуля  
 {  
 puts("| Continent | Country | Max density | Country | Min density |");  
 puts("-----------------------------------------------------------------------------");  
 printf("|%14s |%14s |%12ld |%14s |%12ld |\n", cntnt, max, maxd, min, mind); //вывод  
 puts("-----------------------------------------------------------------------------");  
 }  
 else //иначе такого континента нет в списке  
 {  
 printf("There is no such continent in the list\n");  
 }  
 getchar();  
}  
  
list new\_node (list begin, DataType country)  
{  
 list temp = (list) malloc(sizeof(struct node)); //выделение памяти под узел  
 temp->data = country; //присваивание значения узлу  
 temp->next = begin; //перенос указателя в начало списка  
 return temp;  
}  
  
list delete\_node (list begin)  
{  
 struct node \*temp;  
 if (begin == NULL)  
 {  
 puts("List is empty");  
 getchar();  
 }  
 if (begin)  
 {  
 temp = begin;  
 begin = begin->next;  
 free(temp);  
 puts("Deleted");  
 getchar();  
 }  
 return begin;  
}

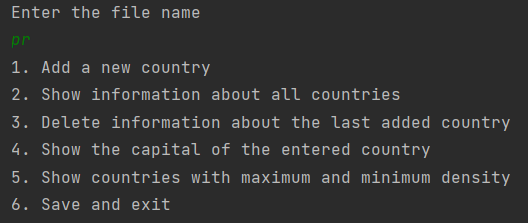
Результаты работы программы:

При запуске программы нужно ввести название файла.

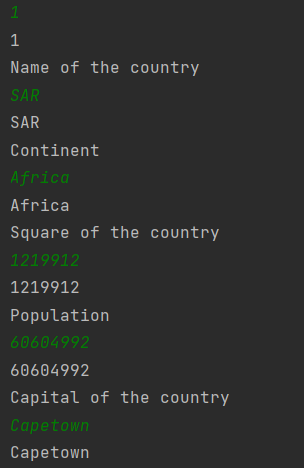
Если файла не существует, то появится ошибка, что файла нет в директории и сообщение о том, что файл был создан, а также появится меню



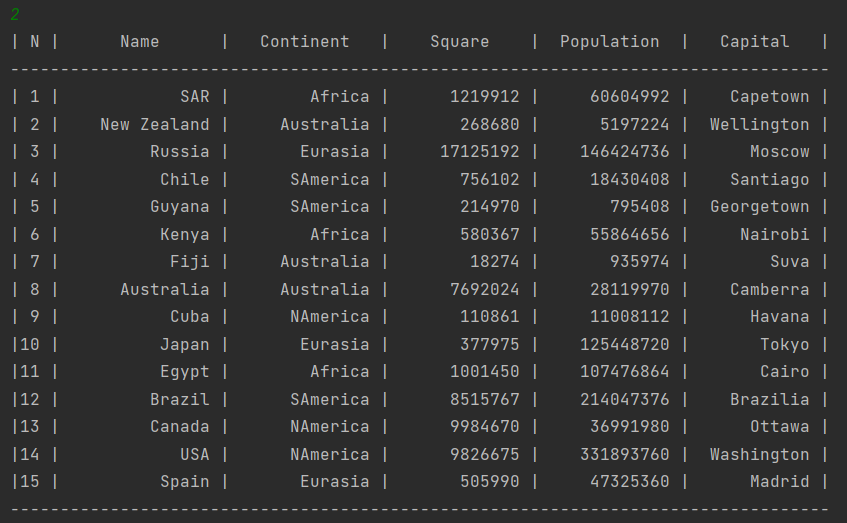
Если файл существует, то меню появится сразу



При нажатии на «1» можно ввести новые данные о стране. С клавиатуры вводится название страны, континент, площадь, население и столица



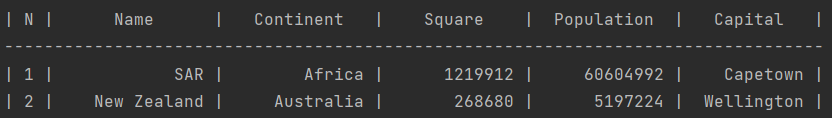
При нажатии на «2» выводятся все имеющиеся данные



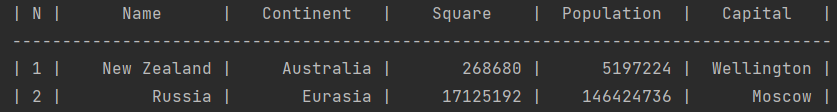
Если же файл еще пуст, то выведется сообщение об этом



При нажатии на «3» удаляется информация о последней добавленной стране



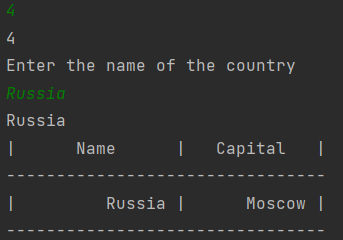




Если файл пуст, то при попытке удаления выведется сообщение об этом



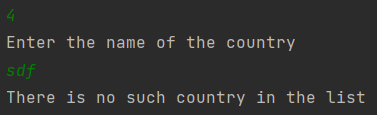
При нажатии на «4» будет выведена столица введенной с клавиатуры страны при условии, что такая имеется в списке



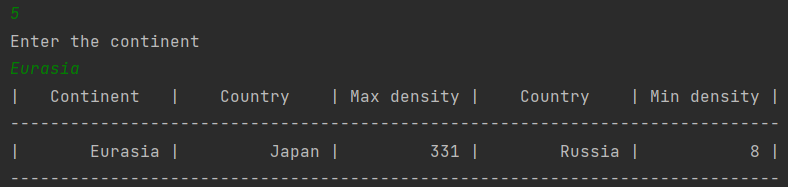
Если файл пуст, то выведется сообщение об этом



Если ввести название страны, которой нет в списке, выведется сообщение об этом



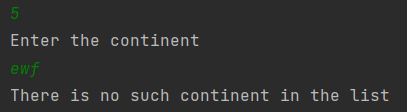
При нажатии на «5» будут выведены названия стран с наибольшей и наименьшей плотностью населения и сама плотность, находящиеся на континенте, введенном с клавиатуры и при условии, что континент и страны есть в списке



Если список пуст, выведется сообщение об этом



Если ввести название страны, которого нет в списке, выведется сообщение об этом



При нажатии на «6» файл будет сохранен, и программа завершится

