Министерство найки и высшего образования РФ
Федеральное государственное автономное образовательное
учреждение высшего профессионального образования
«Санкт-Петербургский государственный электротехнический
университет «ЛЭТИ» им. В.И.Ульянова(Ленина)»
(СПБГЭТУ «ЛЭТИ»)

Факультет компьютерных технологий и информатики

Кафедра вычислительной техники

Отчёт к курсовой работе на тему «Разработка электронной картотеки» по дисциплине «Программирование»

Выполнил: студент гр.0306 Семёнов М.Д.

Проверила: к.т.н.,доцент Сискович Т.И.

Санкт-Петербург
2021

Содержание

Введение

Целью выполнения курсовой работы № 2 является получение практических навыков в использовании основных элементов языка программирования.

1. Задание

Создать электронную картотеку, хранящуюся на диске, и программу, обеспечивающую взаимодействие с ней.

Программа должна выполнять следующие действия:

- занесение данных в электронную картотеку;
- внесение изменений (исключение, корректировка, добавление);
- поиск данных по различным признакам;
- сортировку по различным признакам;
- вывод результатов на экран и сохранение на диске.

2. Уточнение задания

Выберем в качестве предметной области список автомобилей.

В программме будет как и основное меню, так и подменю для некоторых пунктов.

Занесение данных в картотеку сделаем как с клавиатуры, так и из файла, причём ввод из файла будет посимвольным. Хранить данные будем в массиве структур.

Внесение каких-либо изменений будем производить только с клавиатуры.

Сделаем возможным перезаписать некторый элемент массива структур, сортировку по любому полю,

удаление какого-либо элемента и добавление элементов.

Вывод как в консоль будем осуществлять в виде таблицы, для более удобного восприятия информации пользователем, в файл будем записывать данные в таком же формате,в каком и получали данные из файла.

3.Описание спроектированных структур

Описание спроектированных структур, использованных в программе приведено в таблице 1.

Таблица 1.Описание спроектированных структур

Имя	Тип	Описание
model	char*	Модель
strana	char*	Страна
massa	int	Масса
year	int	Год выпуска
speed	float	Скорость
mochnost	float	Мощность
h_l_w	float	Характеристики

4.Контрольные примеры

Контрольные примеры представлены в таблице 1.

№	Файл test.txt	Заданное значение	Действие	Выходные данные
1	Toyota 1;Japan;2004;1090;205;125;8.9.1 Mercedes 1;Germany;2006;1590;220;130;9.11.2 Mercedes 2;Germany;2008;2090;240;240;139.11.2 Mercedes 3;Germany;2006;2090;250;97;129.11.2 Mercedes 4;Germany;2011;1900;300;97;117.11.2 Toyota 2;Japan;2002;1855;172;107;8.9.3 Mercedes 5;Germany;2004;1900;205;97;99.12.4 Lada 2009;Russia;2006;2500;190;115;8.9.3 Toyota 001;Camry;2016;1650;210;102;8.9.3 Mitsubisi 179;China;2012;2050;205;143;8.9.3	1	Удалить	| № | Модель | Страна | Масса| Год |Скорость|Мощность| Характеристики | +----+----------+---------+------+------+--------+--------+----------+ | 1 | Mercedes 1 | Germany | 2006 | 1590 | 220,00 | 130,00 | 9,00 | 11,00 | 2,00 | | 2 | Mercedes 2 | Germany | 2008 | 2090 | 240,00 | 240,00 | 139,00 | 11,00 | 2,00 | | 3 | Mercedes 3 | Germany | 2006 | 2090 | 250,00 | 97,00 | 129,00 | 11,00 | 2,00 | | 4 | Mercedes 4 | Germany | 2011 | 1900 | 300,00 | 97,00 | 117,00 | 11,00 | 2,00 | | 5 | Toyota 2 | Japan | 2002 | 1855 | 172,00 | 107,00 | 8,00 | 9,00 | 3,00 | | 6 | Mercedes 5 | Germany | 2004 | 1900 | 205,00 | 97,00 | 99,00 | 12,00 | 4,00 | | 7 | Lada 2009 | Russia | 2006 | 2500 | 190,00 | 115,00 | 8,00 | 9,00 | 3,00 | | 8 | Toyota 001 | Camry | 2016 | 1650 | 210,00 | 102,00 | 8,00 | 9,00 | 3,00 | | 9 | Mitsubisi 179 | China | 2012 | 2050 | 205,00 | 143,00 | 8,00 | 9,00 | 3,00 |
2		Lada 2009	Поиск	| № | Модель | Страна | Масса| Год |Скорость|Мощность| Характеристики | +----+----------+---------+------+------+--------+--------+----------+ | 1 | Lada 2009 | Russia | 2006 | 2500 | 190,00 | 115,00 | 8,00 | 9,00 | 3,00 |
3		Germany	Поиск	| № | Модель | Страна | Масса| Год |Скорость|Мощность| Характеристики | +----+----------+---------+------+------+--------+--------+----------+ | 1 | Mercedes 1 | Germany | 2006 | 1590 | 220,00 | 130,00 | 9,00 | 11,00 | 2,00 | | 2 | Mercedes 2 | Germany | 2008 | 2090 | 240,00 | 240,00 | 139,00 | 11,00 | 2,00 | | 3 | Mercedes 3 | Germany | 2006 | 2090 | 250,00 | 97,00 | 129,00 | 11,00 | 2,00 | | 4 | Mercedes 4 | Germany | 2011 | 1900 | 300,00 | 97,00 | 117,00 | 11,00 | 2,00 | | 5 | Mercedes 5 | Germany | 2004 | 1900 | 205,00 | 97,00 | 99,00 | 12,00 | 4,00 |

5.Краткое описание алгоритма.

Начало

Шаг 1.Пользователю открывается меня,в котором есть пункты:

1.Ввести список.Список вводится либо из файла либо из консоли.

2.Удаление элементов. Удаление элемента происходит в соответствии с номером,который ввёл пользователь.

3.Добавление элементов. Элементы добавляются в конец списка.

4.Корректировка элементов. Пользователь указывает номер элемента,который хочет отредактировать. Вводит новые значения элемента

5.Поиск данных по различным признакам. Пользователь указывает данные по которым он хочет найти элементы. В соответствии с введённым значением формируется новый список, состоящий из элементов с указанными полями структур.

6.Сортировка по различным признакам. Список сортируется по признаку,который указал пользователь.

7.Вывод результатов в консоль или в файл. Результат записывается в файл,либо выводится в консоль.

8.Выйти . Завершение работы программы.

Конец

6.Структура вызова функций

Структура вызова функций, используемых в программе , приведена на Рис.1.

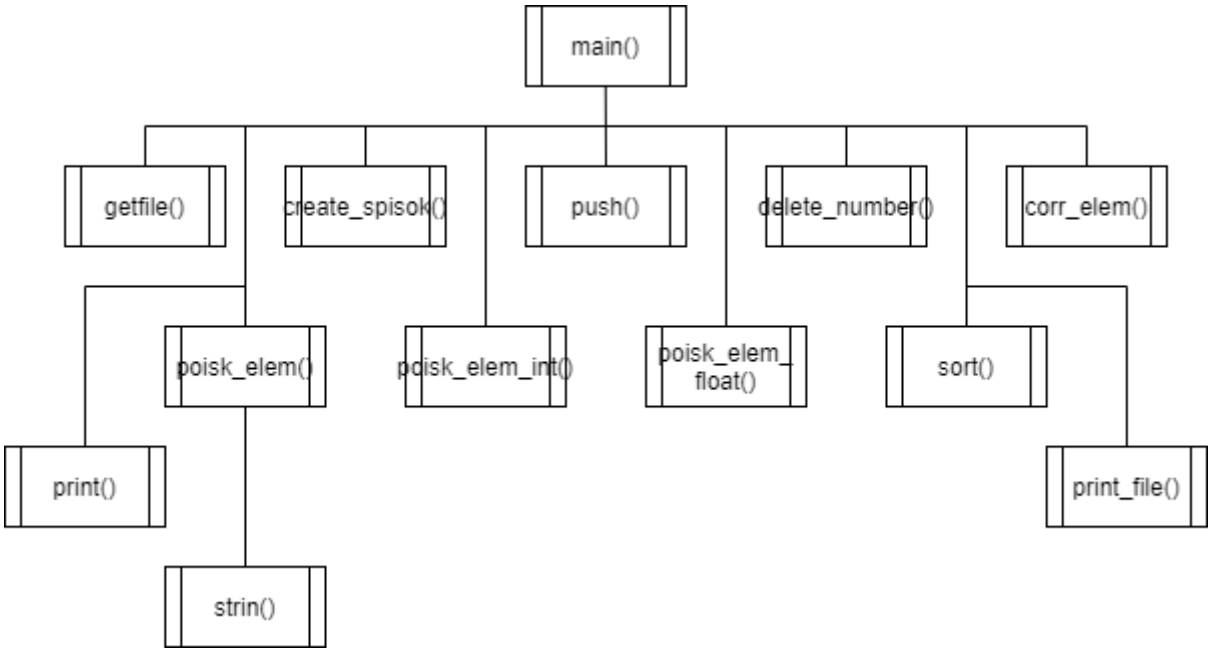


Рис.1.Структура вызова функций

7.Функции

7.1Функция main()

Назначение — является основной логической функцией в данной программе,которая вызывает другие функции.

Функция возвращает 0.

Функция ничего не принимает на вход.

Прототип int main()

Описание переменных функции приведено в Таблице 3.

Схема функции представлена на Рис.2.

Таблица 3.Описание переменных функции main

№	Имя	Тип	Описание
1	list	Spisok*	Голова списка
2	tmp	Spisok*	Текущий элемент списка
3	file	FILE*	Имя файла
4	input	int	Переменная выбора
5	k	int	Проверка на 1ую структуру списка
6	flag	int	Переменная выбора
7	i	int	Счётчик
8	n	int	Количество структур,вводимых из консоли

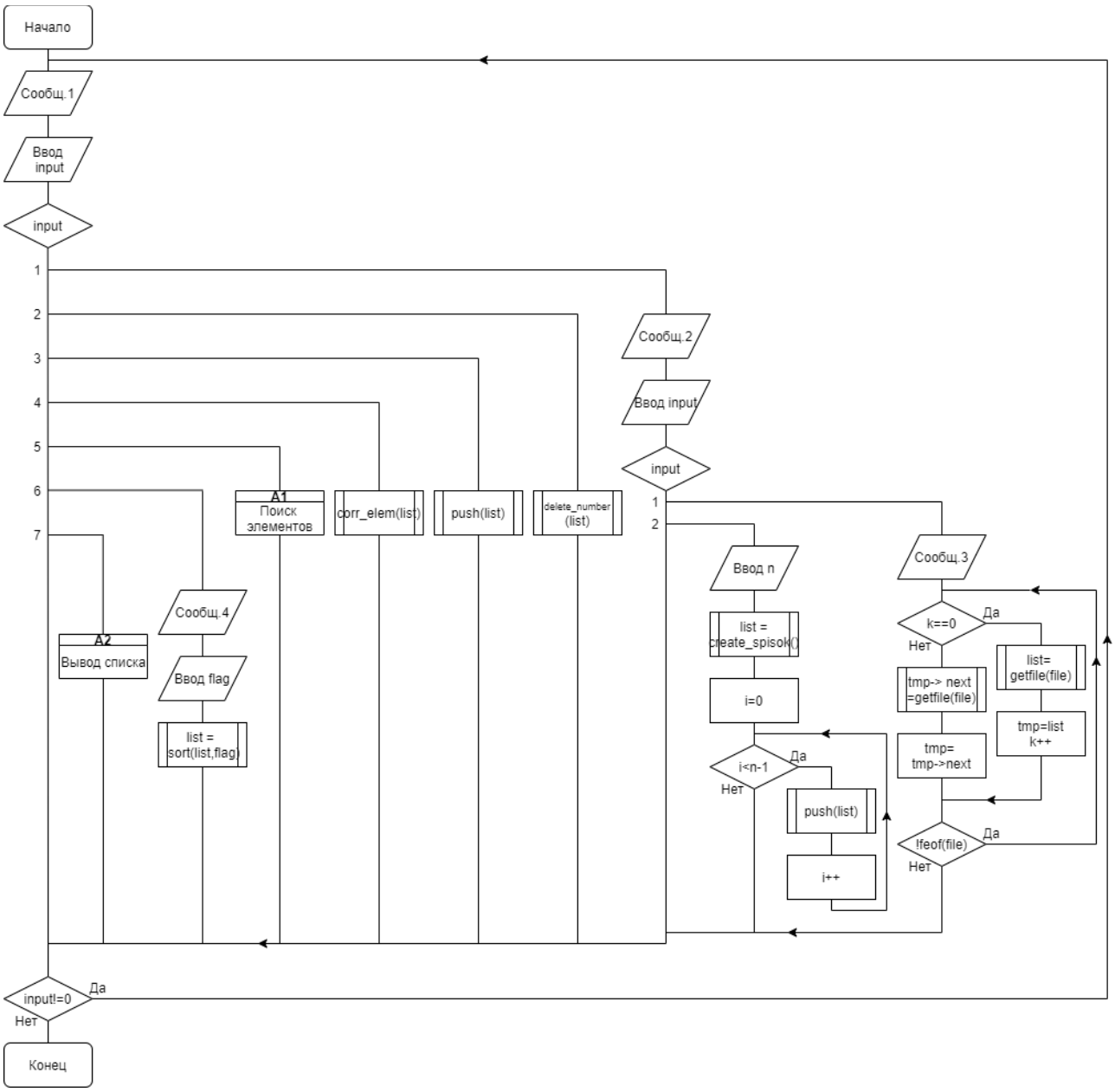


Рис.1.Схема функции main

7.2 Функция create_spisok()

Назначение — Создание списка в консоли.

Функция возвращает голову списка.

Функция ничего не принимает на вход.

Прототип Spisok* create_spisok()

Описание переменных функции приведено в Таблице 4.

Таблица 4.Описание переменных функции create_spisok

№	Имя	Тип	Описание
1	str	char*	Строка
2	slen	int	Длина строки
3	i	int	Счётчик

7.3 Функция getfile()

Назначение — Чтение структур из файла.

Функция возвращает структуру.

Функция принимает на вход имя файла.

Прототип Spisok* getfile(FILE*)

Описание переменных функции приведено в Таблице 5.

Таблица 5.Описание переменных функции getfile()

№	Имя	Тип	Описание
1	file	FILE*	Параметр:имя файла
2	tmp	char*	Строка
3	I,j	int	Счётчик
4	str	char*	Строка
5	ch	char	Символ-разделитель

7.4Функция **strin()**

Назначение —выясняет равны ли 2 строки .

Функция возвращает 1 — если строки равны, 0 -если не равны.

Функция принимает на вход 2 строки и длину первой строки.

Прототип int strin(char* ,char*,int)

Описание переменных функции приведено в Таблице 6.

Таблица 6.Описание переменных функции strin

№	Имя	Тип	Описание
1	**str**	char*	Параметр:строка
2	**str2**	char*	Параметр:строка
3	slen	int	Параметр:длина строки
4	flag	int	Возвращаемое значение
5	i	int	Счётчик

7.5**Функция poisk_elem()**

Назначение —Поиск элементов по информационному полю .

Функция возвращает голову списка.

Функция принимает на вход голову списка.

Прототип Spisok* poisk_elem(Spisok*)

Описание переменных функции приведено в Таблице 7.

Таблица 7.Описание переменных функции poisk_elem

№	Имя	Тип	Описание
1	plist	Spisok*	Параметр:голова списка
2	var	int	Выбор пользователя
3	str	char*	Строка
4	str2	char*	Строка
5	c	int	Результат вызываемой функции
6	p	Spisok*	Текущий элемент

7.6Функция poisk_elem_int()

Назначение —Поиск элементов по числовому полю .

Функция возвращает голову списка.

Функция принимает на вход голову списка.

Прототип poisk_elem_int(Spisok*)

Описание переменных функции приведено в Таблице 8.

Таблица 8.Описание переменных функции poisk_elem_int

№	Имя	Тип	Описание
1	plist	Spisok*	Параметр:голова списка
2	var	int	Выбор пользователя
3	cislo	float	Ввод пользователя
4	c	int	Результат вызываемой функции
5	p	Spisok*	Текущий элемент

7.7Функция corr_elem()

Назначение —Корректировка элемента.

Функция ничего не возвращает .

Функция принимает на вход голову списка.

Прототип void corr_elem(Spisok *plist)

Описание переменных функции приведено в Таблице 9.

Таблица 9.Описание переменных функции corr_elem

№	Имя	Тип	Описание
1	plist	Spisok*	Параметр:голова списка
2	var	int	Выбор пользователя
3	q	Spisok*	Вспомог
4	new_element	Spisok*	Новый узел
5	p	Spisok*	Текущий элемент
6	i	int	Счётчик

7.8Функция delete_number()

Назначение —Удаление элемента.

Функция ничего не возвращает .

Функция принимает на вход голову списка.

Прототип void delete_number(Spisok* plist)

Описание переменных функции приведено в Таблице 10.

Таблица 10.Описание переменных функции delete_number

№	Имя	Тип	Описание
1	plist	Spisok*	Параметр:голова списка
2	var	int	Выбор пользователя
3	q	Spisok*	Вспомог
4	p	Spisok*	Текущий элемент
5	i	int	Счётчик

7.9Функция push()

Назначение —Добавление элемента.

Функция ничего не возвращает .

Функция принимает на вход голову списка.

Прототип void push(Spisok *plist)

Описание переменных функции приведено в Таблице 11.

Таблица 11.Описание переменных функции push

№	Имя	Тип	Описание
1	plist	Spisok*	Параметр:голова списка
2	*new_element	Spisok*	Новый элемент
3	p	Spisok*	Текущий элемент

7.10 Функция sort()

Назначение —Сортировка списка.

Функция возвращает голову списка.

Функция принимает на вход голову списка,выбор пользователя.

Прототип sort(Spisok*,int).

Описание переменных функции приведено в Таблице 12.

Таблица 12.Описание переменных функции sort

№	Имя	Тип	Описание
1	head	Spisok*	Параметр:голова списка
2	flag	int	Параметр:выбор пользователя
3	A,c,d	Spisok*	Вспомогательные элементы
4	cur	Spisok*	Текущий элемент
5	X,key	int	Вспомогательные элементы

7.11 Функция print()

Назначение —Вывод списка в консоль.

Функция ничего не возвращает .

Функция принимает на вход голову списка.

Прототип void print(Spisok *).

Описание переменных функции приведено в Таблице 13.

Таблица 13.Описание переменных функции print

№	Имя	Тип	Описание
1	list	Spisok*	Параметр:голова списка
2	t	Spisok*	Текущий элемент
3	i	int	Счётчик

7.12 Функция print_file()

Назначение —Вывод списка в файл.

Функция ничего не возвращает .

Функция принимает на вход голову списка.

Прототип void print_file(Spisok *).

Описание переменных функции приведено в Таблице 14.

Таблица 14.Описание переменных функции print_file

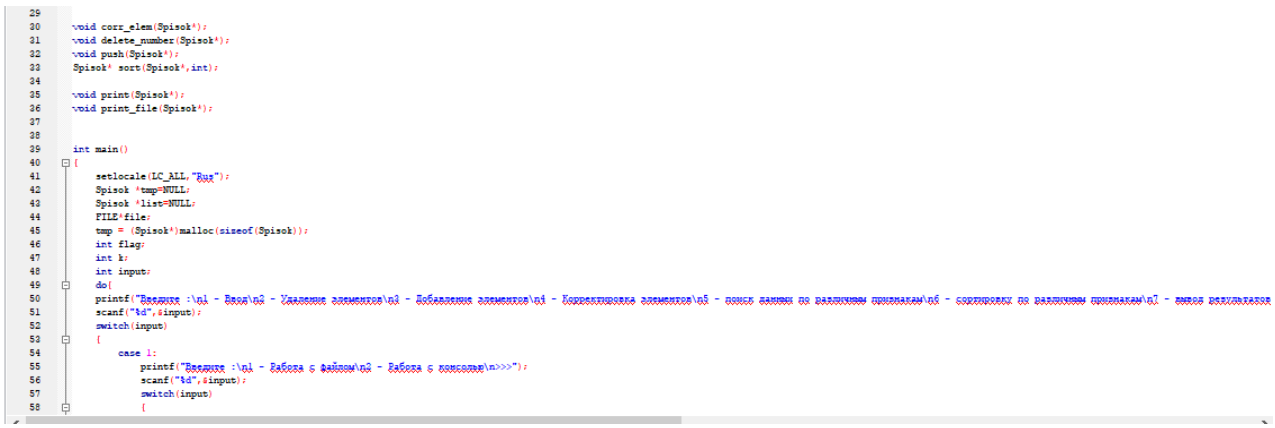
№	Имя	Тип	Описание
1	plist	Spisok*	Параметр:голова списка
2	cur	Spisok*	Текущий элемент
3	FILE*	file	Имя файла
4	i	int	Счётчик

8.Текст программы

Текст программы приведён на рисунках 5-

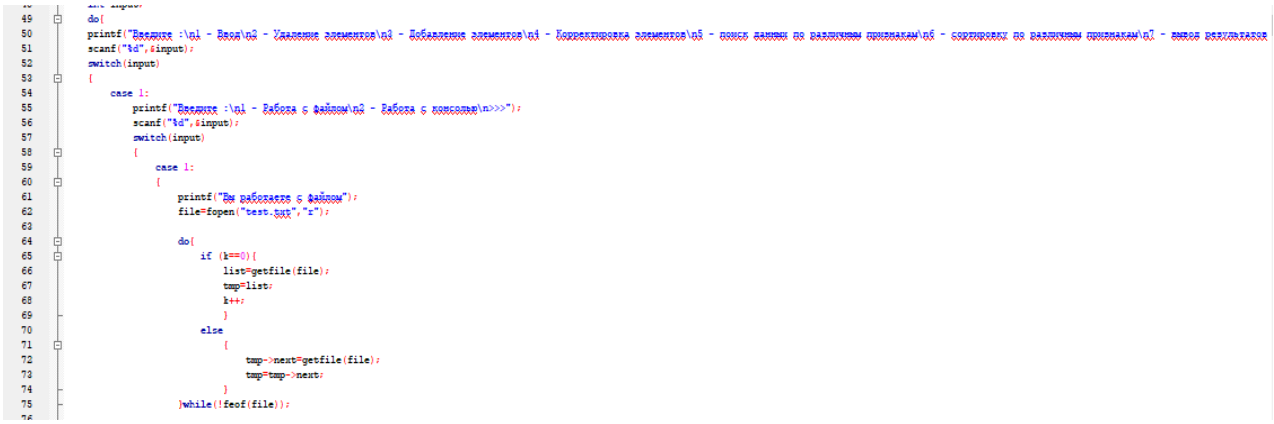
```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <locale.h>


typedef struct Node
{

char* model;
char* strana;
int year;
int massa;
float speed;
float mochnost;
float h_l_w[3];
struct Node *next;

} Spisok;


Spisok* create_spisok();
Spisok* getfile(FILE*);

int strin(char*,char*,int);
Spisok* poisk_elem(Spisok*);
Spisok* poisk_elem_int(Spisok*);
Spisok* poisk_elem_float(Spisok*);

```

Рис.5.Текст программы(1)



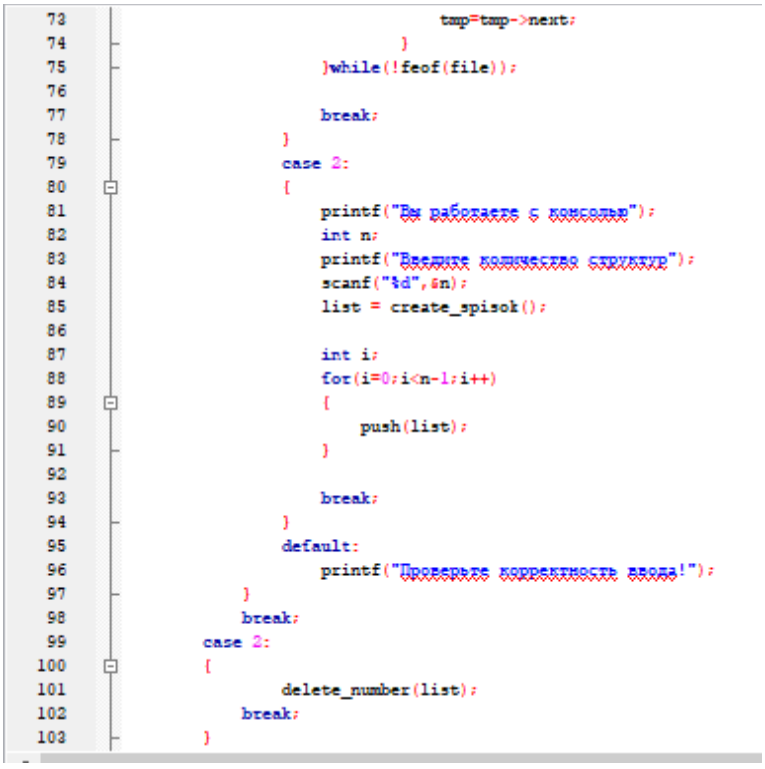
```
29
30  void corr_elem(Spisok*);
31  void delete_number(Spisok*);
32  void push(Spisok*);
33  Spisok* sort(Spisok*,int);
34
35  void print(Spisok*);
36  void print_file(Spisok*);
37
38
39  int main()
40  {
41      setlocale(LC_ALL,"Rus");
42      Spisok *tmp=NULL;
43      Spisok *list=NULL;
44      FILE*file;
45      tmp = (Spisok*)malloc(sizeof(Spisok));
46      int flag;
47      int k;
48      int input;
49      do{
50      printf("Введите :\n1 - Ввод\n2 - Удаление элементов\n3 - Добавление элементов\n4 - Корректировка элементов\n5 - поиск данных по различным признакам\n6 - сортировку по различным признакам\n7 - вывод результатов
51      scanf("%d",&input);
52      switch(input)
53      {
54          case 1:
55              printf("Введите :\n1 - Работа с файлом\n2 - Работа с консолью\n>>>");
56              scanf("%d",&input);
57              switch(input)
58              {
```

Рис.6.Текст программы(2)



```
49      do{
50      printf("Введите :\n1 - Ввод\n2 - Удаление элементов\n3 - Добавление элементов\n4 - Корректировка элементов\n5 - поиск данных по различным признакам\n6 - сортировку по различным признакам\n7 - вывод результатов
51      scanf("%d",&input);
52      switch(input)
53      {
54          case 1:
55              printf("Введите :\n1 - Работа с файлом\n2 - Работа с консолью\n>>>");
56              scanf("%d",&input);
57              switch(input)
58              {
59                  case 1:
60                  {
61                      printf("Вы работаете с файлом");
62                      file=fopen("test.txt","r");
63
64                      do{
65                          if (k==0){
66                              list=getfile(file);
67                              tmp=list;
68                              k++;
69                              }
70                          else
71                              {
72                                  tmp->next=getfile(file);
73                                  tmp=tmp->next;
74                              }
75                      }while(!feof(file));
```

Рис.7.Текст программы(3)



```
73                                  tmp=tmp->next;
74                              }
75                      }while(!feof(file));
76
77                      break;
78                  }
79                  case 2:
80                  {
81                      printf("Вы работаете с консолью");
82                      int n;
83                      printf("Введите количество структур");
84                      scanf("%d",&n);
85                      list = create_spisok();
86
87                      int i;
88                      for(i=0;i<n-1;i++)
89                      {
90                          push(list);
91                      }
92
93                      break;
94                  }
95                  default:
96                      printf("Проверьте корректность ввода!");
97              }
98              break;
99          case 2:
100         {
101                 delete_number(list);
102             break;
103         }
```

Рис.8.Текст программы(4)

```
        }
        case 3:
        {
            push(list);
            break;
        }
        case 4:
        {
            corr_elem(list);
            break;
        }
        case 5:
        {

            printf("1 - Информационное поле\n2 - Целочисленное поле\n3 - Вещественное поле\n>>>");
            scanf("%d",&input);
            switch(input)
            {
            case 1:
                list = poisk_elem(list);
                break;
            case 2:
                list = poisk_elem_int(list);
                break;
            case 3:
                list = poisk_elem_float(list);
               break;
            default:
                printf("Проверьте корректность ввода!");
            }
            break;
```

Рис.9.Текст программы(5)

```
            break;
    }

        case 6:
        {

            printf("Введите поле\n>>>");
            scanf("%d",&flag);
            if (flag<10){
            list = sort(list,flag);
            }
            else{
                printf("Проверьте корректность ввода!");
            }
            break;
        }
        case 7:
        {
            printf("1 - В консоль\n2 - В файл\n>>>");
            scanf("%d",&input);
            switch(input)
            {
            case 1:
                print(list);
                break;
            case 2:
                print_file(list);
            break;
            default:
                printf("Проверьте корректность ввода!");
            }
```

Рис.10.Текст программы(6)

```
163             }
164             break;
165         }
166             default:
167                 printf("Проверьте правильность ввода!");
168     }
169     }while(input!=0);
170     return 0;
171 }
172
173 Spisok* create_spisok()
174 {
175     char* str=NULL;
176     int i,slen;
177     Spisok * node=(Spisok*)malloc(sizeof(Spisok));
178     str=malloc(128*sizeof(char));
179     getchar();
180     for (i=0;i<2;i++)
181     {
182     if(i==0)
183         printf("Введите модель");
184     if(i==1)
185         printf("Введите страну");
186     fgets(str,128,stdin);
187     slen=strlen(str);
188     str[slen-1]='\0';
189     if (i==0){
190     node->model=(char*)malloc(slen*sizeof(char));
191     strcpy(node->model,str);
192     }
193     else{
```

Рис.11.Текст программы(7)

```
193     else{
194     node->strana=(char*)malloc(slen*sizeof(char));
195     strcpy(node->strana,str);
196     }
197     }
198     printf("Введите массу\n>>>");
199     scanf("%d",&node->massa);
200     printf("Введите год\n>>>");
201     scanf("%d",&node->year);
202     printf("Введите скорость\n>>>");
203     scanf("%f",&node->speed);
204     printf("Введите мощность\n>>>");
205     scanf("%f",&node->mochnost);
206     for(i=0;i<3;i++)
207     {
208     printf("Введите значение массива\n>>>");
209     scanf("%f",&node->h_l_w[i]);
210     }
211     node->next=NULL;
212     return node;
213 }
214
215 Spisok* getfile(FILE* file)
216 {
217     Spisok* tmp=malloc(sizeof(Spisok));
218     char ch;
219     int i,j;
220     char* str=NULL;
221     j=0;
222     for(i=0;i<2;i++)
223     {
```

Рис.12.Текст программы(8)

```
223     {
224     str=malloc(128*sizeof(char));
225     while((ch=fgetc(file))!=';')
226     {
227         str[j]=ch;
228         j++;
229     }
230     str=realloc(str,(j+1)*sizeof(char));
231     str[j]='\0';
232     if (i==0)
233     {
234         tmp->model=malloc((j+1)*sizeof(char));
235         strcpy(tmp->model,str);
236     }
237     if (i==1)
238     {
239         tmp->strana=malloc((j+1)*sizeof(char));
240         strcpy(tmp->strana,str);
241     }
242     free(str);
243     str=NULL;
244     j=0;
245     }
246     fscanf(file,"%d;%d;%f;%f;%f.%f.%f*c",&tmp->massa,&tmp->year,&tmp->speed,&tmp->mochnost,&tmp->h_l_w[0],&tmp->h_l_w[1],&tmp->h_l_w[2]);
247     ch=fgetc(file);
248     tmp->next=NULL;
249     return tmp;
250 }
251
252
```

Рис.13.Текст программы(9)

```
253 int strin(char* str,char*str2,int slen)
254 {int flag;
255     flag=1;
256     int i;
257
258     for(i=0;i<slen&&(flag==1);i++)
259     {
260         if (str2[i]!=str[i])
261         {
262             flag=0;
263         }
264
265     }
266     return flag;
267 }
268
269 Spisok* poisk_elem(Spisok* plist)
270 {
271     int var,slen,c;
272     c=0;
273     char* str,*str2;
274     Spisok* p;
275     p=plist;
276     getchar();
277         printf("\n1 - Модель\n2 - Страна\n>>>");
278         scanf("%d",&var);
279         if(var==1){
280             str=malloc(128*sizeof(char));
281             getchar();
282             fgets(str,128,stdin);
283             slen=strlen(str);
```

Рис.14.Текст программы(10)

```
283             slen=strlen(str);
284             str[slen-1]='\0';
285             while(p!=NULL && c==0)
286             {
287                 str2=malloc((strlen(p->model))*sizeof(char));
288                 strcpy(str2,p->model);
289                 c=strin(str,str2,slen);
290                 if(c==0)
291                     p=p->next;
292                 str2=NULL;
293             }
294             plist=p;
295             while(p->next!=NULL)
296             {
297                 str2=malloc((strlen(p->next->model))*sizeof(char));
298                 strcpy(str2,p->next->model);
299                 c=strin(str,str2,slen);
300                 if(c==1)
301                 {
302                     p=p->next;
303                 }
304                 else{
305                     p->next=p->next->next;
306                 }
307             }
308         }
309         else{
310
311             str=malloc(128*sizeof(char));
312             getchar();
313             fgets(str,128,stdin);
```

Рис.15.Текст программы(11)

```
310
311             str=malloc(128*sizeof(char));
312             getchar();
313             fgets(str,128,stdin);
314             slen=strlen(str);
315             str[slen-1]='\0';
316             while(p!=NULL && c==0)
317             {
318                 str2=malloc((strlen(p->strana))*sizeof(char));
319                 strcpy(str2,p->strana);
320                 c=strin(str,str2,slen);
321                 printf("%d\n",c);
322                 if(c==0)
323                     p=p->next;
324                 str2=NULL;
325             }
326             plist=p;
327             while(p->next!=NULL)
328             {
329                 str2=malloc((strlen(p->next->strana))*sizeof(char));
330                 strcpy(str2,p->next->strana);
331                 c=strin(str,str2,slen);
332                 if(c==1)
333                 {
334                     p=p->next;
335                 }
336                 else{
337                     p->next=p->next->next;
338                 }
339             }
340         }
```

Рис.16.Текст программы(12)

```
340         }
341
342     return plist;
343 }
344
345
346 Spisok* poisk_elem_int(Spisok*plist)
347 {
348     int var;
349     int cislo;
350     Spisok* p;
351     p=plist;
352         printf("\n1 - Год\n2 - Модель\n>>>");
353         scanf("%d",&var);
354         printf("\nВведите значение>>>");
355         scanf("%d",&cislo);
356         switch(var)
357         {
358         case 1:
359             while(p!=NULL && p->year!=cislo)
360             {
361                 if(p->year!=cislo)
362                     p=p->next;
363             }
364             plist=p;
365             if (p!=NULL)
366             {
367             while(p->next!=NULL)
368             {
369                 if(p->next->year==cislo)
370                 {
```

Рис.17.Текст программы(13)

```
370                 {
371                     p=p->next;
372                 }
373                 else{
374                     p->next=p->next->next;
375                 }
376             }
377             }
378             else {
379                 printf("Элементы не найдены!");
380             }
381         break;
382         case 2:
383            while(p!=NULL && p->massa!=cislo)
384             {
385                 if(p->massa!=cislo)
386                     p=p->next;
387             }
388             plist=p;
389             if (p!=NULL)
390             {
391             while(p->next!=NULL)
392             {
393                 if(p->next->massa==cislo)
394                 {
395                     p=p->next;
396                     printf("&\n");
397                 }
398                 else{
399                     p->next=p->next->next;
400                     printf("#\n");
```

Рис.18.Текст программы(14)

```
400                     printf("#\n");
401                 }
402             }
403             }
404         break;
405         default:
406             printf("Проверьте корректность ввода!");
407         }
408         return plist;
409 }
410
411 Spisok* poisk_elem_float(Spisok*plist)
412 {
413     int var,i,count;
414     float cislo;
415     i=0;
416     Spisok* p;
417     p=plist;
418         printf("\n1 - Скорость\n2 - Мощность\n3 - Характеристики\n>>>");
419         scanf("%d",&var);
420         printf("\nВведите значение>>>");
421         scanf("%f",&cislo);
422         switch(var)
423         {
424         case 1:
425             while(p!=NULL && p->massa!=cislo)
426             {
427                 if(p->massa!=cislo)
428                 {
429                     p=p->next;
430                 }
```

Рис.19.Текст программы(15)

```
430             }
431         }
432         if (p!=NULL)
433         {
434         plist=p;
435         while(p->next!=NULL)
436         {
437             if(p->next->massa==cislo)
438             {
439                 p=p->next;
440             }
441             else{
442                 p->next=p->next->next;
443             }
444         }
445         }
446         else{
447             printf("Элементы не найдены!");
448         }
449      break;
450      case 2:
451         while(p!=NULL && p->massa!=cislo)
452         {
453             if(p->massa!=cislo)
454                 p=p->next;
455         }
456         if(p!=NULL)
457         {
458         plist=p;
459         while(p->next!=NULL)
460         {
```

Рис.20.Текст программы(16)

```
460             {
461                 if(p->next->massa==cislo)
462                 {
463                     p=p->next;
464                 }
465                 else{
466                     p->next=p->next->next;
467                 }
468             }
469             }
470             else{
471                 printf("Элементы не найдены!");
472             }
473         break;
474         case 3:
475             printf("\n1 - 1ый элемент\n1 - 2ой элемент\n2 - 3ий элемент");
476             scanf("%d",&count);
477 
478             while(p!=NULL && p->h_l_w[count]!=cislo)
479             {
480                 if(p->h_l_w[count]!=cislo)
481                     p=p->next;
482             }
483             plist=p;
484 
485             while(p->next!=NULL)
486             {
487                 if(p->h_l_w[count]==cislo)
488                 {
489                     p=p->next;
490                 }
```

Рис.21.Текст программы(17)

```
490                 }
491                 else{
492                     p->next=p->next->next;
493                 }
494             }
495 
496 
497         break;
498         }
499         return plist;
500 }
501 
502 
503 void corr_elem(Spisok *plist)
504 {
505     Spisok* p;
506     Spisok* q;
507     Spisok *new_element= NULL;
508     p=plist;
509     int var,i;
510     i=1;
511     printf("Введите номер списка,который хотите поменять\n>>>");
512     scanf("%d",&var);
513     if (var==1)
514     {
515         new_element=create_spisok();
516         new_element->next=p->next;
517         plist=new_element;
518     }
519     else{
520     while(i+2<=var)
521     {
522         p=p->next;
523         i++;
```

Рис.22.Текст программы(18)

```
523             i++;
524
525         }
526         new_element=create_spisok();
527         q=p->next;
528         new_element->next=p->next->next;
529         p->next=new_element;
530         free(q->model);
531         free(q->strana);
532         q->model=NULL;
533         q->strana=NULL;
534         q=NULL;
535         }
536     }
537
538
539
540
541 void delete_number(Spisok* plist)
542 {
543     int i,var;
544     Spisok * p;
545     Spisok * q;
546     p=plist;
547     printf("Введите номер,который хотите удалить\n>>>");
548     scanf("%d",&var);
549     if (var==1)
550     {
551         p=plist->next;
552         free(plist->model);
553         plist->model=NULL;
554         free(plist->strana);
555         plist->strana=NULL;
556         free(plist);
```

Рис.23.Текст программы(19)

```
547     printf("Введите номер,который хотите удалить\n>>>");
548     scanf("%d",&var);
549     if (var==1)
550     {
551         p=plist->next;
552         free(plist->model);
553         plist->model=NULL;
554         free(plist->strana);
555         plist->strana=NULL;
556         free(plist);
557         plist=NULL;
558         plist=p;
559     }
560     else{
561         while(i+2<var)
562         {
563             p=p->next;
564             i++;
565         }
566             q=p->next;
567             p->next=p->next->next;
568             free(q);
569         q=NULL;
570     }
571 }
572
573
574
575 void push(Spisok *plist)
576 {
577     Spisok *new_element= NULL;
578     Spisok * p;
579     p=plist;
580     while(p->next!=NULL)
```

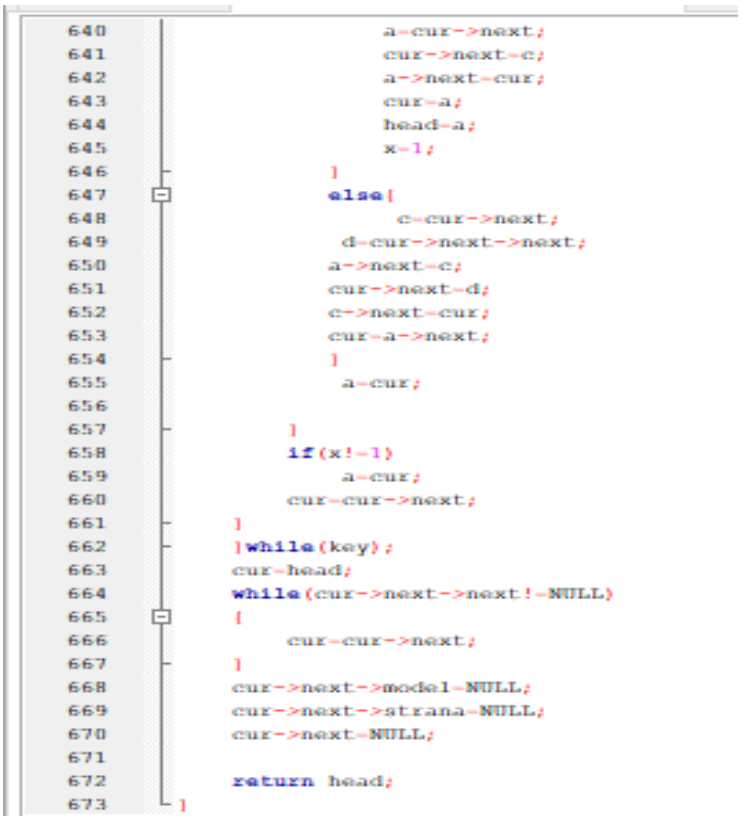
Рис.24.Текст программы(20)

```
577         Spisok *new_element= NULL;
578         Spisok * p;
579         p=plist;
580         while(p->next!=NULL)
581             p=p->next;
582             new_element=create_spisok();
583         p->next=new_element;
584     }
585
586     Spisok* sort(Spisok* head,int flag)
587     {
588         Spisok* cur,*a=NULL,*c,*d;
589         int key;
590         int tmp=0;
591         int x=0;
592         cur=head;
593         while(cur!=NULL)
594         {
595             c=cur;
596             cur=cur->next;
597         }
598         cur=(Spisok*)malloc(sizeof(Spisok));
599         /*cur->model="0";
```

Рис.25.Текст программы(21)

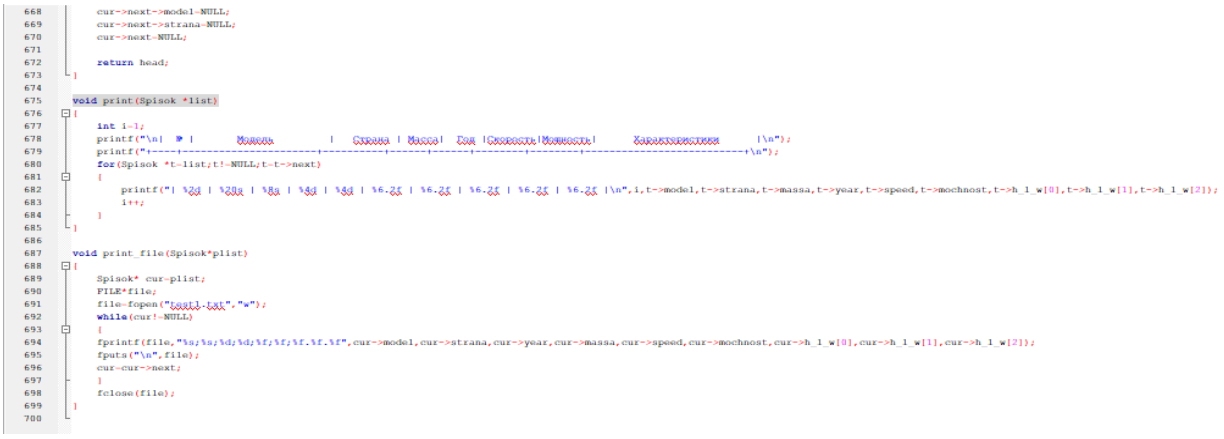
```
609     c->next=cur;
610     cur->next=NULL;
611
612
613
614     do{
615             key = 0;
616             a=NULL;
617         cur=head;
618     while(cur->next->next!=NULL)
619     {
620         if (flag==1 )
621         {
622             tmp=strcmp(cur->model,cur->next->model);
623         }
624         if (flag==2 )
625         {
626             tmp=strcmp(cur->strana,cur->next->strana);
627         }
628         x=0;
629
630         if((flag==4 && cur->massa>cur->next->massa)||(flag==3 && cur->year>cur->next->year)||(flag==5 && cur->speed>cur->next->speed)||(flag==6 && cur->mochnost>cur->next->mochnost)||
631            (flag==7 && cur->h_l_w[0]>cur->next->h_l_w[0])||(flag==8 && cur->h_l_w[1]>cur->next->h_l_w[1])||(flag==9 && cur->h_l_w[2]>cur->next->h_l_w[2])
632            ||(flag==1 && tmp==1)||(flag==2 && tmp==1)
633            )
634         {
635             key=1;
636            if (a==NULL)
637            {
638                a=(Spisok*)sizeof(Spisok);
639                c=cur->next->next;
640                a=cur->next;
```

Рис.26.Текст программы(22)



```
640                     a=cur->next;
641                     cur->next=c;
642                     a->next=cur;
643                     cur=a;
644                     head=a;
645                     x=1;
646                 }
647                 else{
648                       c=cur->next;
649                  d=cur->next->next;
650                 a->next=c;
651                 cur->next=d;
652                 c->next=cur;
653                 cur=a->next;
654                 }
655                  a=cur;
656
657             }
658             if(x!=1)
659                  a=cur;
660             cur=cur->next;
661         }
662         }while(key);
663         cur=head;
664         while(cur->next->next!=NULL)
665         {
666             cur=cur->next;
667         }
668         cur->next->model=NULL;
669         cur->next->strana=NULL;
670         cur->next=NULL;
671
672         return head;
673     }
```

Рис.27.Текст программы(23)

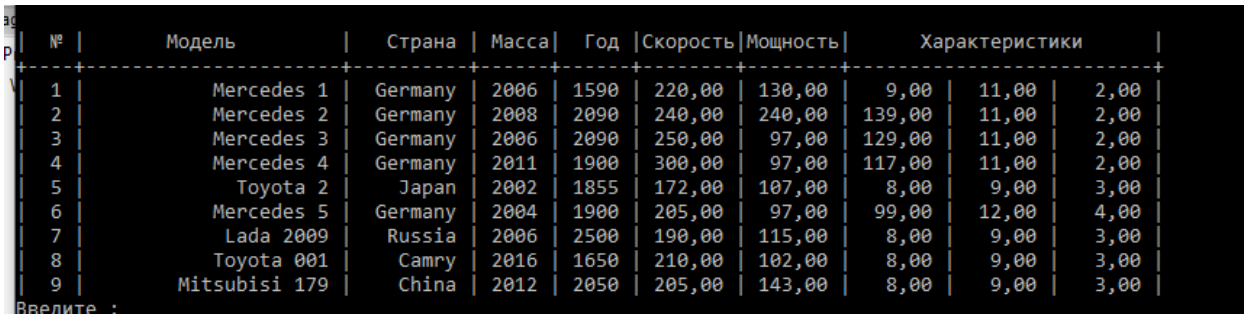


```
668     cur->next->model=NULL;
669     cur->next->strana=NULL;
670     cur->next=NULL;
671
672     return head;
673 }
674
675 void print(Spisok *list)
676 {
677     int i=1;
678     printf("\n|  №  |       Модель        |   Страна  | Масса|  Год  |Скорость|Мощность|      Характеристики      |\n");
679     printf("+-----+---------------------+-----------+------+-------+--------+--------+--------------------------+\n");
680     for(Spisok *t=list;t!=NULL;t=t->next)
681     {
682         printf("| %2d | %20s | %8s | %4d | %4d | %6.2f | %6.2f | %6.2f | %6.2f | %6.2f |\n",i,t->model,t->strana,t->massa,t->year,t->speed,t->mochnost,t->h_l_w[0],t->h_l_w[1],t->h_l_w[2]);
683         i++;
684     }
685 }
686
687 void print_file(Spisok*plist)
688 {
689     Spisok* cur=plist;
690     FILE*file;
691     file=fopen("testl.txt","w");
692     while(cur!=NULL)
693     {
694     fprintf(file,"%s;%s;%d;%d;%f;%f;%f.%f.%f",cur->model,cur->strana,cur->year,cur->massa,cur->speed,cur->mochnost,cur->h_l_w[0],cur->h_l_w[1],cur->h_l_w[2]);
695     fputs("\n",file);
696     cur=cur->next;
697     }
698     fclose(file);
699 }
700
```

Рис.28.Текст программы(24)

9.Результаты выполнения программы

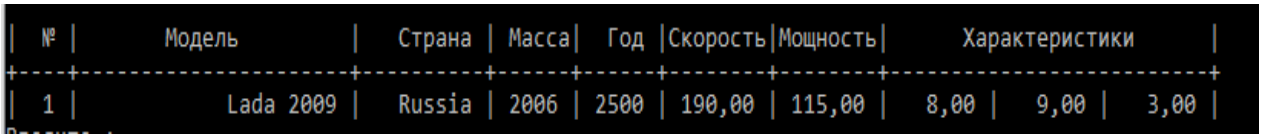
При выполнении программы получены результаты ,совпадающие с ожидаемыми в Таблице 2. Ошибки не обнаружены. Результаты выполнения программы согласно контрольным примерам.



№	Модель	Страна	Масса	Год	Скорость	Мощность	Характеристики		
1	Mercedes 1	Germany	2006	1590	220,00	130,00	9,00	11,00	2,00
2	Mercedes 2	Germany	2008	2090	240,00	240,00	139,00	11,00	2,00
3	Mercedes 3	Germany	2006	2090	250,00	97,00	129,00	11,00	2,00
4	Mercedes 4	Germany	2011	1900	300,00	97,00	117,00	11,00	2,00
5	Toyota 2	Japan	2002	1855	172,00	107,00	8,00	9,00	3,00
6	Mercedes 5	Germany	2004	1900	205,00	97,00	99,00	12,00	4,00
7	Lada 2009	Russia	2006	2500	190,00	115,00	8,00	9,00	3,00
8	Toyota 001	Camry	2016	1650	210,00	102,00	8,00	9,00	3,00
9	Mitsubisi 179	China	2012	2050	205,00	143,00	8,00	9,00	3,00

Введите :

Рис.29.Результаты выполнения программы(1)



№	Модель	Страна	Масса	Год	Скорость	Мощность	Характеристики		
1	Lada 2009	Russia	2006	2500	190,00	115,00	8,00	9,00	3,00

Рис.30.Результаты выполнения программы(2)

№	Модель	Страна	Масса	Год	Скорость	Мощность	Характеристики		
1	Mercedes 1	Germany	2006	1590	220,00	130,00	9,00	11,00	2,00
2	Mercedes 2	Germany	2008	2090	240,00	240,00	139,00	11,00	2,00
3	Mercedes 3	Germany	2006	2090	250,00	97,00	129,00	11,00	2,00
4	Mercedes 4	Germany	2011	1900	300,00	97,00	117,00	11,00	2,00
5	Mercedes 5	Germany	2004	1900	205,00	97,00	99,00	12,00	4,00

Введите :

Рис.31.Результаты выполнения программы(3)

Заключение

В результате выполнения курсовой работы были закреплены навыки в использовании списквв ,а также была создана электронная картотека.