**Федеральное государственное автономное образовательное учреждение**

**высшего профессионального образования**

**"Национальный исследовательский университет**

**"Высшая школа экономики"**

Московский институт электроники и математики НИУ ВШЭ  
Департамент компьютерной инженерии

**Курс: Информатика и программирование**

**ОТЧЕТ**  
**по лабораторной работе № 4**

|  |  |  |  |
| --- | --- | --- | --- |
| **Раздел** | **Max оценка** | **Итог.**  **Оценка**  **1** | **Итог.**  **Оценка**  **2** |
| Тесты | 1 |  |  |
| Работа программы | 1 |  |  |
| Алгоритм | 3 |  |  |
| Вопросы | 2 |  |  |
| Доп. Задание | 3 |  |  |

**Студент: Жалкова Наталия Евгеньевна**  
**Группа: БИВ153, 1 подгруппа**   
**Вариант: 76**  
**Руководитель: Хруслова Диана Владимировна  
Оценка:**

**МОСКВА 2016**

# Содержание

[Содержание 1](#_Toc443325017)

[Условия задач 2](#_Toc443325018)

[Распечатка программы 3](#_Toc443325019)

[Тестовые примеры 6](#_Toc443325020)

# Условия задач

Ввод и вывод данных, а также формирование массива результатов оформить как отдельные функции. Проверку существования результата произвести в главной программе. К элементам массива структур обращаться при помощи индекса в первой программе и при помощи указателя во второй программе. При сравнении строк использовать стандартные функции для обработки символьных строк. Для ввода данных и вывода результатов использовать функции scanf, gets и printf, puts в первой программе и функции fscanf, fgets. fputs и fprintf во второй программе.

Написать программу, которая вводит в режиме запросов заданное число структур вида: 

I. После ввода массива структур программа ищет в нем следующую информацию:

33. Модели, у которых имеется максимальное количество различных объемов двигателя.

II. Модифицировать программу таким образом, чтобы она вводила структуры из текстового файла и записывала результаты в файл.

# Распечатка программы

**1 программа**

#include <stdio.h>

#include <string.h>

#include <cstdlib> //для atoi

#include <ctype.h> //для isdigit

using namespace std;

#define CLR while (getchar()!='\n')

//очистка входной строки

//вложенная структура (1)

struct model

{

int seriya,number;

};

//вложенная структура (2)

struct date

{

int year, month;

};

struct car

{model mod; //вложенная структура (1)

date dat; //вложенная структура (2)

char firm[20], color[20];

int price, engine;

};

void proverka(int \*prov)

{int k;

char sprov[20],\*s;

do

{k=0; //количество цифр

gets(sprov); //ввод строки

s=sprov; //указатель на строку

while(\*s) //пока существует символ

{if (isdigit(\*s)) //если это цифра, счетчик увеличивается

k++;

s++;}//идем дальше

if (k!=strlen(sprov)) //если количество цифр не равно длине строки

printf("Enter positive integer\r\n");

}

while (k!=strlen(sprov)); //пока количество цифр не равно длине строки вводим строку снова и снова

\*prov=atoi(sprov); //вышли из цикла, ввод верен, привели строку к типу int

}

//ввод

void read(int \*kol, car cat[])

{\*kol=0;

int i,prov;

do

{printf("Enter number of cars\r\n");

scanf("%d",kol);

if (\*kol<0||\*kol>200)

printf("Wrong number, please repeat\r\n");

}

while(\*kol<0||\*kol>200);

CLR;

while (i<\*kol)

{printf("\r\nCar %d\r\n",++i);

printf("Firm........");

while (\*gets(cat->firm)==NULL);

printf("Model...Series.....");

proverka(&prov);

cat->mod.seriya=prov;

printf("Model...Number.....");

proverka(&prov);

cat->mod.number=prov;

printf("Price........");

do

{proverka(&prov);

if (prov==0)

printf("Enter positive integer\r\n");}

while (prov<=0);

cat->price=prov;

printf("Release date...Year.....");

do

{proverka(&prov);

if (prov>2016||prov<1769)

printf("Repeat enter 1768<year<2017\r\n");}

while (prov>2016||prov<1769);

cat->dat.year=prov;

printf("Release date...Month.....");

do

{proverka(&prov);

if (prov<1||prov>12)

printf("Repeat enter 0<month<13\r\n");}

while (prov<1||prov>12);

cat->dat.month=prov;

printf("Color........");

while (\*gets(cat->color)==NULL);

printf("Engine capacity........");

proverka(&prov);

cat->engine=prov;

cat++;}

}

//Модели, у которых имеется максимальное количество различных объемов двигателя.

void search(car cat[], int kol, int mod[], int \*n1)

{

int i,j,diff=0,max=0;

\*n1=-1;//изначально нет моделей

if (kol==1)

{\*n1=0;

mod[\*n1]=0;

}

else

{for(i=0;i<kol-1;i++)//высчитываем максимум разных обьемов

{diff=0;

for (j=1;j<kol;j++)

{if (cat[i].mod.seriya==cat[j].mod.seriya&&cat[i].mod.number==cat[j].mod.number)//если одна модель

if (cat[i].engine!=cat[j].engine)//если у них разные двигатели

diff++;//учитываем эту разность

}

if (diff>max) max=diff; //вычисляем максимум

}

for(i=0;i<kol;i++) //находим какие модели имеют этот максимум

{diff=0;

for (j=i+1;j<kol;j++)

{if (cat[i].mod.seriya==cat[j].mod.seriya&&cat[i].mod.number==cat[j].mod.number)

if (cat[i].engine!=cat[j].engine)

diff++;

}

if (diff==max)//если нашли модель, запоминаем ее

{\*n1=\*n1+1;

mod[\*n1]=i;}

}

}

}

//вывод

void print(int n1, int mod[], car cat[])

{int c,i;

for(i=0;i<n1+1;i++)

{c=mod[i];

if (cat[c].mod.seriya!=cat[c-1].mod.seriya||i==0)

{printf("Model...Series.....%d\r\n",cat[c].mod.seriya);

printf("Model...Number.....%d\r\n\r\n",cat[c].mod.number);

}

}

}

int main (){

int kol,n1;

car catalog[200];

int models[20];

read(&kol,catalog);//ввод

search(catalog, kol, models, &n1);//поиск

if(n1!=-1) //если модели найдены

{printf("\r\nModels with max amount of different engine capacity:");

print(n1, models,catalog);//вывод на экран

}

else

puts("No needed cars were found");

return 0;

}

**2 программа**

#include <stdio.h>

#include <string.h>

#include <ctype.h> //для isdigit

using namespace std;

#define FCLR while (getc(fin)!='\n')

//очистка входной строки

//вложенная структура (1)

struct model

{

int seriya,number;

};

//вложенная структура (2)

struct date

{

int year, month;

};

struct car

{model mod; //вложенная структура (1)

date dat; //вложенная структура (2)

char firm[20], color[20];

int price, engine;

};

//замена в строке символа '\n' на '\0'

void deln(char \*s)

{ for (; \*s && \*s!='\n'; s++);

\*s='\0';

}

//ввод

void read(int \*kol, car cat[], char \*filename)

{FILE \*fin;

car \*ucar;

\*kol=0;

if (!(fin=fopen(filename, "r")))

printf("File %s does not exist\n",filename);

else

{

//во входном файле нет пустых строк и каждый пункт занимает новую строку

for (ucar=cat; !feof(fin); (\*kol)++, ucar++)

if(fgets(ucar->firm,20,fin))

{

deln(ucar->firm);

fscanf(fin,"%d", &(ucar->mod.seriya));

fscanf(fin,"%d", &(ucar->mod.number));

fscanf(fin,"%d", &(ucar->price));

fscanf(fin,"%d", &(ucar->dat.year));

fscanf(fin,"%d", &(ucar->dat.month));

if (!feof(fin)) FCLR;

fgets(ucar->color, 20, fin);

deln(ucar->color);

fscanf(fin,"%d", &(ucar->engine));

//Очистка входного потока

if (!feof(fin)) FCLR;

}

fclose(fin);

}

}

//Модели, у которых имеется максимальное количество различных объемов двигателя.

void search(car cat[], int kol, int mod[], int \*n1)

{car \*iukaz,\*jukaz;

int i,c,diff=0,max=0;

\*n1=-1;//изначально нет моделей

if (kol==1)

{\*n1=0;

mod[\*n1]=0;

}

else

{for(iukaz=cat;iukaz<cat+kol-1;iukaz++)//высчитываем максимум разных обьемов

{diff=0;

for (jukaz=cat;jukaz<cat+kol;jukaz++)

{if (iukaz->mod.seriya==jukaz->mod.seriya&&iukaz->mod.number==jukaz->mod.number)//если одна модель

if (iukaz->engine!=jukaz->engine)//если у них разные двигатели

diff++;//учитываем эту разность

}

if (diff>max) max=diff; //вычисляем максимум

}

for(iukaz=cat,i=0;iukaz<cat+kol;iukaz++,i++) //находим какие модели имеют этот максимум

{diff=0;

for (jukaz=iukaz;jukaz<cat+kol;jukaz++)

{if (iukaz->mod.seriya==jukaz->mod.seriya&&iukaz->mod.number==jukaz->mod.number)

if (iukaz->engine!=jukaz->engine)

diff++;

}

if (diff==max)//если нашли модель, запоминаем ее

{\*n1=\*n1+1;

mod[\*n1]=i;}

}

}

}

//вывод в файл

void print(int n1, int mod[], car cat[], char \*filename)

{int c,i;

FILE \*fout;

fout=fopen(filename, "w");

fprintf(fout,"Models with max amount of different engine capacity:\r\n");

for(i=0;i<n1+1;i++)

{c=mod[i];

if (cat[c].mod.seriya!=cat[c-1].mod.seriya||i==0)

{fprintf(fout,"Model...Series.....%d\r\n",cat[c].mod.seriya);

fprintf(fout,"Model...Number.....%d\r\n\r\n",cat[c].mod.number);

}

}

fclose(fout);

}

int main (){

int kol,n1;

car catalog[200];

int models[20];

FILE \*fout;

read(&kol,catalog,"input.txt");

search(catalog, kol, models, &n1);

if(n1!=-1)

print(n1, models,catalog,"output.txt");

else

{fout=fopen("output.txt", "w");

fprintf(fout,"No needed cars were found");

}

return 0;

}

# Тестовые примеры

|  |  |  |
| --- | --- | --- |
| **№** | **Исходные данные** | **Результат** |
| **1** |  |  |
| **2** |  |  |
| **3** |  |  |
| **4** |  |  |