

1

```
/*
4.
Для хранения данных о планшетных сканерах описать структуру
вида:

struct scan_info
{
    char model [25]; // наименование модели
    int price; // цена
    double x_size; // горизонтальный размер области сканирования
    double y_size; // вертикальный размер области сканирования
    int optr; // оптическое разрешение
    int grey; // число градаций серого
};
```

Написать функцию, которая запрашивает количество сканеров, информация о которых будет вводиться, динамически выделяет память под массив структур соответствующего размера и заполняет его данными в режиме диалога (с клавиатуры). При этом имя сканера может содержать пробелы.

Написать функцию, которая записывает данный массив в создаваемый бинарный файл. Если цена сканера меньше 200, то данные об этом сканере в файл не записываются. Информация об остальных сканерах помещается в бинарный файл, причем сначала пишутся данные о всех сканерах, имя которых начинается с заглавной буквы, а затем – с прописной. Структура файла: в первых четырех байтах размещается значение типа `int`, определяющее количество сделанных в файл записей; далее без пропусков размещаются записи о сканерах.

Привести пример программы, создающей файл с данными о сканерах и осуществляющий вывод на дисплей данных о требуемой записи (либо всех, либо по номеру). Все необходимые данные для функций должны передаваться им в качестве параметров.

Использование глобальных переменных в функциях не допускается.

```
MSP,
центр новых информационных технологий "Северная Звезда"
гимназии "Лаборатория Салахова",
20.10.09, 16:32,
21.10.09, 09:00–22:52;
*/
```

```
38 # include <stdio.h>
39 # include <conio.h>
40 # include <stdlib.h> // malloc ()
41 # include <ctype.h> // islower ()
42 # include <process.h> // exit ()
```

```

43     struct scan_info
44     {
45         char model [25]; // наименование модели
46         int price; // цена
47         double x_size; // горизонтальный размер области сканирования
48         double y_size; // вертикальный размер области сканирования
49         int optr; // оптическое разрешение
50         int grey; // число градаций серого
51     };

52     void write (
53         int N, // число сканеров
54         struct scan_info * A // адрес массива структур
55     )
56     {
57         // MSP, 21.10.09, 11:03
58         //
59         // Массив о сканерах записывается в бинарный файл.
60         //
61         // Структура файла: в первых четырех байтах размещается значение
62         // типа int, определяющее количество сделанных в файл записей;
63         // далее без пропусков размещаются записи о сканерах.
64         //
65         // Если цена сканера меньше 200,
66         // то данные об этом сканере в файл не записываются.
67         // Информация о сканерах помещается в файл таким образом, что
68         // сначала пишутся данные о всех сканерах, имя которых
69         // начинается с заглавной буквы, а затем - с прописной.

70     FILE * F;

71     int
72     N1, // число сканеров, цена которых >= 200
73     i; // номер сканера

74     struct scan_info t; // элемент массива A

75     F = fopen ("D:\\BC\\BIN\\MSP\\scanners.bin", "wb");
76     if (F == NULL)
77         printf ("не открывается файл на запись...\n");

78     // сколько записей будет записано в файл (4 bytes)
79     N1 = N;
80     for (i = 0; i < N; i ++)
81     {
82         //printf ("***   A [%i].price = %i\n", i, A [i].price);
83         if (A [i].price < 200)
84             -- N1;
85     }

```

```

86     if (N1 == 0)
87     {
88         printf ("\nБинарный файл пуст...\n");
89         fclose (F);
90         free (A);
91         getch ();
92         exit (1);
93     }

94     // N1 - столько записей будет в файле
95     // Запишем это число в бинарный файл
96     fprintf (F, "%4i", N1);

97     // Запишем в файл информацию о тех сканерах,
98     // название которых начинается с заглавной буквы
99     for (i = 0; i < N; i ++)
100    {
101        if (
102            A [i].price >= 200
103            &&
104            islower (A [i].model [0]) == 0 // верхний регистр
105        )
106        {
107            t = A [i];
108            // запись в файл структуры
109            fwrite (& t, sizeof (t), 1, F);
110        }
111    }

112    // Допишем в файл информацию о тех сканерах,
113    // название которых начинается с прописной буквы
114    for (i = 0; i < N; i ++)
115    {
116        if (
117            A [i].price >= 200
118            &&
119            islower (A [i].model [0]) != 0 // нижний регистр
120        )
121        {
122            t = A [i];
123            // запись в файл структуры
124            fwrite (& t, sizeof (t), 1, F);
125        }
126    }

127    // закроем файл
128    fclose (F);
129 } // end write (int, struct scan_info)

130 struct scan_info * input (

```

```

131             int * N // адрес числа сканеров
132         )
133     {
134         // MSP, 20.10.09, 16:35
135         //
136         // Запросить количество сканеров,
137         // информация о которых будет вводиться;
138         // динамически выделить память под массив структур
139         // соответствующего размера;
140         // заполнить его данными в режиме диалога (с клавиатуры).
141         // Имя сканера может содержать пробелы.

142         int
143             d, // длина
144             //с, // код символа
145             i, // номер сканера
146             S; // счетчик длины имени сканера

147         double t; // для scanf () при чтении double

148         struct scan_info * A; // адрес массива структур

149         printf ("Сколько сканеров: ");
150         scanf ("%i", N);
151         fflush (stdin);

152         A = (struct scan_info *)
153             malloc (* N * sizeof (struct scan_info));

154         for (i = 0; i < * N; i ++)
155         {
156             // наименование модели
157             printf ("\nНаименование модели %i-го сканера: ", i + 1);
158             gets (A [i].model);
159             //printf ("%s\n", A [i].model);

160             // цена
161             printf ("Цена: ");
162             scanf ("%i", & A [i].price);
163             fflush (stdin);

164             //printf ("price = %i\n", A [i].price);

165             // горизонтальный размер области сканирования
166             printf ("Горизонтальный размер области сканирования: ");
167             scanf ("%lf", & t);
168             fflush (stdin);
169             A [i].x_size = t;
170             //printf ("x_size = %lf\n", A [i].x_size);

171             // вертикальный размер области сканирования

```

```

172     printf ("Вертикальный размер области сканирования: ");
173     scanf ("%lf", & t);
174     fflush (stdin);
175     A [i].y_size = t;
176     //printf ("y_size = %lf\n", A [i].y_size);

177     // оптическое разрешение
178     printf ("Оптическое разрешение: ");
179     scanf ("%i", & A [i].optr);
180     fflush (stdin);
181     //printf ("optr = %i\n", A [i].optr);

182     // число градаций серого
183     printf ("Оптическое разрешение: ");
184     scanf ("%i", & A [i].grey);
185     fflush (stdin);
186     //printf ("grey = %i\n", A [i].grey);
187 }

188     return A;
189 } // end void input (int *)

190 int main ()
191 {
192     FILE * F; // файл о сканерах

193     struct scan_info
194     t, // значение одной из записанных структур
195     * A; // адрес массива структур

196     int
197     i, // номер записи
198     ns, // номер сканера, номер записи в файле
199     N; // число сканеров

200     clrscr ();

201     // заполнение массива
202     A = input (& N);

203     // запись данных о сканерах в бинарный файл
204     write (N, A);

205     F = fopen ("D:\\BC\\BIN\\MSP\\scanners.bin", "rb");
206     if (F == NULL)
207         printf ("не открывается файл на чтение...\n");

208     // прочтем из файла первые 4 байта
209     fscanf (F, "%4i", & N);
210     printf ("\nЧисло записей в файле: %i\n", N);

```

```

211 // вывод на дисплей данных о требуемой записи
212 // (либо всех, либо по номеру).
213 printf ("\nМЕНЮ:\n");
214 printf ("- Вывести информацию о всех сканерах - <0>\n"
215         "- по номеру записи - <номер>");
216 printf ("\nЧто: ");
217 scanf ("%i", & ns);
218 fflush (stdin);
219 printf ("\n");

220 if (ns == 0)
221 {
222     printf ("%25s %6s %8s %8s %4s %4s\n",
223            "Наименование модели", "Цена", "ГРОС", "ВРОС",
224            "ОР", "ЧГС");

225     for (i = 0; i < N; i ++)
226     {
227         // прочтем значение записи о сканере из бинарного файла
228         fread (& t, sizeof (t), 1, F);
229         printf ("%25s %6i %8.2lf %8.2lf %4i %4i\n",
230                t.model, t.price, t.x_size, t.y_size, t.optr,
231                t.grey);
232     }
233 }
234 else
235 {
236     printf ("Данные о записи номер %i:\n", ns);

237     // установим указатель файла на нужную запись
238     // 4 - число байт файла, потраченное на N
239     fseek (F, 4 + sizeof (t) * (ns - 1), SEEK_SET);

240     // прочтем значение записи о сканере
241     fread (& t, sizeof (t), 1, F);
242     printf ("%25s %6s %8s %8s %4s %4s\n",
243            "Наименование модели", "Цена", "ГРОС", "ВРОС",
244            "ОР", "ЧГС");

245     printf ("%25s %6i %8.2lf %8.2lf %4i %4i\n",
246            t.model, t.price, t.x_size, t.y_size, t.optr, t.grey);
247 }

248 // закрыть файл
249 fclose (F);

250 // вернем память OS
251 free (A);

252 getch ();

```

```
253         return 0;
254     } // end main ()
```

Listing данной задачи опубликован в сети Internet по адресу
<http://www.Best-Listing.ru/color-2-task-142.html>

Sergey Mitrofanov, 26.08.14, 22:22

E-mail: infostar@mail.ru

© <http://www.Best-Listing.ru/>, 2006–2014