

```
1 Program sort_lit;
2 {
```

Задача. [LITMO]

Сортировка чисел в массиве – одна из самых базовых задач в программировании. Разные сортировки отличаются друг от друга требуемым количеством памяти, временем работы, сложностью реализации и некоторыми менее важными параметрами.

Периодически возникают задачи, в которых массив необходимо сортировать, используя некоторую нетривиальную операцию. Одну из таких задач Вам и предлагается решить. В качестве сортирующей операции Вы можете несколько раз использовать разворот последовательности из нескольких подряд идущих первых чисел массива. Например, Вы можете превратить массив (2, 3, 1, 4, 5) в массив (3, 2, 1, 4, 5), развернув первые два элемента, а затем превратить полученный массив в массив (1, 2, 3, 4, 5), развернув в первые три элемента.

Формат входного файла

В первой строке входного файла input.txt находится одно натуральное число N ($1 \leq N \leq 20$) – количество элементов в массиве. В следующей строке перечислены сами элементы массива – натуральные числа, не превышающие 1000. Все элементы массива различны.

Формат выходного файла

В первой строке выходного файла output.txt требуется вывести одно целое число t , не превышающее 10000 – количество разворотов, которые Вы собираетесь сделать. В следующей строке необходимо через пробел вывести t натуральных чисел, не превышающих N , где очередное число является количеством первых элементов массива, которые Вы собираетесь развернуть соответствующей операцией. После выполнения выведенных Вами операций массив должен быть отсортирован по возрастанию.

Пример входных и выходных данных

input.txt	output.txt
5	2
2 3 1 4 5	2 3

Решение.

Сергей Митрофанов,
Вячеслав Одинокоев,
гимназия "Лаборатория Салахова", г. Сургут,
12.03.14, 09:49–12:06, DT = 2h17m

```
}
```

```

45     Type mas = array [1..20] of Integer;

46     Var
47     { массив длин разворотов }
48     D : array [1..10000] of Integer;

49     B, { какие числа уже есть в массиве A }
50     A { массив для сортировки }
51     : mas;

52     N, { длина массива }
53     j, { индекс массива длин разворотов }
54     rev, { число разворотов }
55     i_max, { индекс максимального }
56     i_min, { индекс минимального }
57     e, { элемент массива }
58     i, { индекс массива }
59     L, { левая граница подмассива }
60     R { правая граница подмассива }
61     : Integer;

62     Procedure revers (var X : mas; L : Integer; R : Integer);
63     {
64     MSP, 12.03.14, 12:03
65     перевернем отрезок массива с 1-го элемента по L-тый
66     }

67     var
68     t, { для обмена }
69     f, { число обменов }
70     i { номер обмена }
71     : Integer;

72     begin
73     f := (R - L + 1) div 2;

74     for i := 1 to f do
75     begin
76     t := X [L];
77     A [L] := A [R];
78     A [R] := t;
79     Inc (L);
80     Dec (R);
81     end;
82     end;

83     Function imax (X : mas; L : Integer; R : Integer) : Integer;
84     {
85     MSP, 12.03.14, 12:04

```

```

    найдем индекс минимального элемента массива
}

88  var
89    i, { индекс массива }
90    max, { максимальный элемент }
91    i_max { индекс максимального }
92    : Integer;

93  begin
94    max := X [L];
95    i_max := L;
96    for i := L + 1 to R do
97      if X [i] > max
98        then
99          begin
100             max := X [i];
101             i_max := i;
102           end;
103    i_max := i_max;
104  end;

105  Function imin (X : mas; L : Integer; R : Integer) : Integer;
106  {
    MSP, 12.03.14, 12:05
    найдем индекс минимального элемента массива
}

110  var
111    i, { индекс массива }
112    min, { минимальный элемент }
113    i_min { индекс минимального }
114    : Integer;

115  begin
116    min := X [L];
117    i_min := L;
118    for i := L + 1 to R do
119      if X [i] < min
120        then
121          begin
122             min := X [i];
123             i_min := i;
124           end;
125    i_min := i_min;
126  end;

127  Procedure print_om (X : mas; N : Integer);
128  {

```

MSP, 12.03.14, 12:07
печать массива, для отладки кода

```
}  
  
132   var i : Integer;  
  
133   begin  
134     Write ('(');  
135     for i := 1 to N do  
136       begin  
137         Write (X [i]);  
138         if i <> N  
139           then  
140             Write (', ');  
141         end;  
142     WriteLn (')');  
143   end;  
  
144   Begin  
  
145     (*  
146     Этот код оставляем для тех, кто пожелает вводить массив не  
147     через файл, а рандомически.  
  
148     { признаки занятости числа, какие числа уже выбраны Random }  
  
149     for i := 1 to N do  
150       B [i] := 0;  
  
151     {  
152     заполним массив так, чтобы все числа в нем были различны и  
153     встречались ровно по 1 разу и были из интервала [1..N]  
154     }  
155     Randomize;  
156     i := 1;  
157     while i <= N do  
158       begin  
159         e := Random (N) + 1;  
  
160         if B [e] = 0  
161           then  
162             begin  
163               A [i] := e;  
164               Inc (i);  
165               B [e] := 1;  
166             end;  
167         end;  
  
168     print_om (A, N);  
169     *)
```

```

170     Assign (input, 'input.txt');
171     Reset (input);

172     ReadLn (N);

173     for i := 1 to N do
174         Read (A [i]);

175     rev := 0;
176     L := 1;
177     R := N;
178     j := 0;
179     while R - L <> 1 do
180         begin
181             { найдем индекс минимального числа }
182             i_min := imin (A, L, R);
183             {WriteLn ('i_min = ', i_min);}

184             if i_min = L
185                 then
186                     begin
187                         Inc (L);

188                         Continue;
189                     end;

190             {
191                 найдем индекс максимального элемента
192                 в отрезке массива от L до i_min
193             }
194             i_max := imax (A, L, i_min - 1);
195             {WriteLn ('i_max = ', i_max);}

196             {
197                 перевернем отрезок массива A от 1 до i_max
198
199                 максимальный элемент может оказаться
200                 под 1-м (левым) индексом (L)
201             }
201             if i_max <> L
202                 then
203                     begin
204                         revers (A, L, i_max);
205                         Inc (rev);
206                         {print_om (A, N);}
207                         Inc (j);
208                         D [j] := i_max - L + 1;
209                         {WriteLn ('L = ', D [j]);}
210                     end;

```

```

211      { перевернем отрезок массива A от 1 до i_min }
212      revers (A, L, i_min);
213      Inc (rev);
214      Inc (j);
215      D [j] := i_min - L + 1;
216      {WriteLn ('L = ', D [j]);}
217      {print_om (A, N);}

218      { результат: минимальный элемент находится под индексом 1 }
219      Inc (L);

220      {ReadKey;}
221      end;

222      {print_om (A, N);}

223      Assign (output, 'output.txt');
224      Rewrite (output);

225      WriteLn (rev);

226      for i := 1 to j do
227          Write (D [i], ' ');

228      End.

```

Listing данной задачи опубликован в сети Internet по адресу
<http://www.Best-Listing.ru/color-10-task-439.html>

Sergey Mitrofanov, 31.03.14, 13:28

E-mail: infostar@mail.ru

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