

{

TITLE: Олимпиадная информатика

File: euroeng.pas

Задача. Европейская комиссия планирует принять решение о том, что официальным языком Евросоюза станет английский. Был также разработан план упрощения английской письменности, который планируется реализовать за четыре года. Первоочередной задачей будет избавление от буквы *s*, которая в сочетаниях *si* и *se* будет изменяться на *s*, в сочетании *sk* – опускаться, а в остальных случаях заменяться на *k*. При этом все замены будут производиться в строгом порядке слева направо. То есть, например, в слове *success* сначала первая из двух букв *s* заменится на *k*, а затем вторая – на *s*, то есть получится *suksess*. А слово *ssk* превратится в *kk*.

На второй год из английских слов изымут все удвоенные буквы: *ee* изменят на *i*, *oo* – на *u*, а в остальных комбинациях будут просто писать одну букву вместо двух одинаковых. Такие замены также будут делать строго в порядке слева направо. Так, слово *ooo* превратится в *uo*, а *oou* – просто в *u* (в нем сначала *oo* заменится на *u*, а затем *uu* – на *u*), слово *iee* превратится в *i* (в нем сначала *ee* заменится на *i*, а затем *ii* – на *i*).

На третий год на конце слова станут опускать букву *e*, если она не единственная буква в слове.

Наконец, завершением реформы станет отмена артиклей (в английском языке три артикля: *a*, *an* и *the*). При этом удаляться эти артикли будут только тогда, когда они в исходном тексте были словами *a*, *an*, *the*. То есть, например, текст *the table* после реформ первых трех лет превратится в *th tabl*, а после реформы четвертого года – просто в *tabl*. А слово *aaaaa* после реформы первых лет станет словом *a*, но поскольку изначально оно не было словом *a* (артиклем), то оно в итоге так и останется словом *a*.

Требуется написать программу, которая будет переводить классический английский текст на Евроинглиш. Во входном файле *Input.txt* записана одна строка текста, состоящая не более чем из 200 символов: латинских строчных и заглавных букв, пробелов и знаков препинания (в тексте могут встречаться: точка, запятая, вопросительный и восклицательный знаки, двоеточие, тире, точка с запятой, открывающаяся и закрывающаяся скобки, апострофы, кавычки).

Заглавные буквы могут встречаться только в начале слова. Нигде подряд не могут стоять два пробела. В начале и конце строки не может стоять пробел. Слова отделяются друг от друга пробелами и/или знаками препинания.

В выходной файл Output.txt нужно выдать преобразованную строку при ограничениях:

- начинаться с заглавной буквы должны те и только те слова, которые начинались с заглавной буквы в исходном тексте;
- не должно встречаться двух пробелов подряд;
- пробелы между словами и знаками препинания должны остаться там и только там, где они были в исходной строке, в начале и конце строки пробелов быть не должно.

Input.txt	Output.txt
casao and coffe	kakao and kofi
Cinderella! Where Is The Dress???	Sinderela! Wher Is Dres???
'A' is a letter	' ' is letter
!!!Hello!!!A-the-"word"	!!!Helo!!!--"word"
Aaaa then the ckckck	A then k
"A"-the an	""-
A the an	
success	sukses

Решение. Калмыков Вадим (ProCrypt),
г. Сургут, ЦНИТ "Северная Звезда",
23:48, 03.10.2008

Source : <http://acm.dvpion.ru>

Compiler : Borland Pascal 7.0

}

```

87   Program EuroEnglish;
88   Type
89     Str = String [205]; { Объявляем свой тип данных }
90   Var
91     A : Str; { Текст из входного файла }

92   {
    Функция Letter = True, если C является буквой, и false в противном
    случае.
    }

96   function Letter (c : Char) : Boolean;
97   begin
98     c := UpCase (c); { для удобства переводим в верхний регистр }
99     if (c >= 'A')
100        and
101        (c <= 'Z')
102        then
103          Letter := True
104        else
105          Letter := false;
106   end;

107  {
    Процедура DelArticle удаляет все артикли из текста. Метками мы
    пользоваться не будем – плохой стиль программирования. После
    удаления артикля длина строки меняется, но поиск нам нужно
    начинать сначала, поэтому процедура DelArticle должна
    быть рекурсивной.
    }

114  procedure DelArticle (var A : Str);
115  var
116    i : Integer; { счетчик }
117  begin
118    { сначала будем искать артикль a }
119    for i := 2 to (Length (A) - 1) do
120      { a является артиклем, если слева и справа от нее нет букв }
121      if (UpCase (A [i]) = 'A')
122         and
123         not Letter (A [i - 1])
124         and
125         not Letter (A [i + 1])
126      then
127        begin
128          Delete (A, i, 1); { удаляем артикль }

129          { удаляем все пробелы справа от артикля }
130          while (A [i] = ' ')
131             and
132             (i <= Length (A))

```

```

133         do
134             Delete (A, i, 1);

135         { рекурсивно вызываем процедуру }
136         DelArticle (A);
137     end;

138     { теперь ищем артикль an }
139     for i := 2 to (Length (A) - 2) do
140         if (UpCase (A [i]) = 'A')
141             and
142             (UpCase (A [i + 1]) = 'N')
143             and
144             not Letter (A [i - 1])
145             and
146             not Letter (A [i + 2])
147         then
148             begin
149                 Delete (A, i, 2); { удаляем артикль }

150                 { удаляем все пробелы справа от артикля }
151                 while (A [i] = ' ')
152                     and
153                     (i <= Length (A))
154                 do
155                     Delete (A, i, 1);

156                 { рекурсивно вызываем процедуру }
157                 DelArticle (A);
158             end;

159     { теперь ищем артикль the }
160     for i := 2 to (Length (A) - 3) do
161         if (UpCase (A [i]) = 'T')
162             and
163             (UpCase (A [i + 1]) = 'H')
164             and
165             (UpCase (A [i + 2]) = 'E')
166             and
167             not Letter (A [i - 1])
168             and
169             not Letter (A [i + 3])
170         then
171             begin
172                 Delete (A, i, 3); { удаляем артикль }

173                 { удаляем все пробелы справа от артикля }
174                 while (A [i] = ' ')
175                     and
176                     (i <= Length (A))
177                 do

```

```

178         Delete (A, i, 1);

179         { рекурсивно вызываем процедуру }
180         DelArticle (A);
181     end;
182 end;

183 {
    Процедура FYear приводит текст в соответствии с реформами первого
    года. При сочетаниях sk мы не будем удалять с, а будем заменять
    на k, относя этот вариант ко "всем остальным случаям". Когда будем
    избавляться от удвоенных букв, получим тот же эффект.
}

189 procedure FYear (var A : Str);
190 var
191     i : Integer;
192 begin
193     { последнюю букву проверим отдельно }
194     for i := 1 to (Length (A) - 1) do
195         begin
196             if (UpCase (A [i]) = 'C')
197                 then
198                     if (UpCase (A [i + 1]) = 'I')
199                         or
200                         (UpCase (A [i + 1]) = 'E')
201                     then
202                         begin
203                             { учитываем регистр букв }

204                             if (A [i] = 'C')
205                                 then
206                                     A [i] := 'S';

207                             if (A [i] = 'c')
208                                 then
209                                     A [i] := 's';
210                             end
211                         else
212                             begin
213                                 { обязательно учитываем регистр }

214                                 if (A [i] = 'c')
215                                     then
216                                         A [i] := 'k';

217                                 if (A [i] = 'C')
218                                     then
219                                         A [i] := 'K';
220                             end;
221                         end;
222                     end;
223                 end;
224             end;
225         end;
226     end;
227 end;

```

```

222     { проверяем последнюю букву }
223     if (A [Length (A)] = 'c')
224         then
225             A [Length (A)] := 'k';

226         if (A [Length (A)] = 'C')
227             then
228                 A [Length (A)] := 'K'
229     end;

230     {
        Процедура DelDouble удалит все удвоенные буквы из текста согласно
        правилам реформы второго года.
        Удалять будем только вторую букву: например, есть слово Uumп,
        значит, после реформы слово должно начинаться с большой
        буквы – Umп.

        Эта процедура после удаления удвоенной буквы должна продолжать
        работу с начала полученной новой строки, поэтому она реализована
        рекурсивно.
    }

240     procedure DelDouble (var A : Str);
241     var
242         i : Integer; { счетчик }
243     begin
244         for i := 1 to (Length (A) - 1) do
245             begin
246                 if (UpCase (A [i]) = UpCase (A [i + 1]))
247                     and
248                     Letter (A [i])
249                 then
250                     begin
251                         { учитываем регистр букв }
252                         if ((A [i]) = 'e')
253                             then
254                                 A [i] := 'i';

255                                 if ((A [i]) = 'E')
256                                     then
257                                         A [i] := 'I';

258                                 if ((A [i]) = 'o')
259                                     then
260                                         A [i] := 'u';

261                                 if ((A [i]) = 'O')
262                                     then
263                                         A [i] := 'U';

```

```

264         Delete (A, i + 1, 1); { удаляем вторую букву в любом
                                   случае }

266         DelDouble (A); { вызываем рекурсивно }
267     end;
268 end;
269 end;

270 {
    Процедура DelE удаляет все буквы e в конце слов
}

273 procedure DelE (var A : Str);
274 var
275     i : Integer; { счетчик }
276 begin
277     { так как буква e должна стоять в конце слова, то она может быть
        только маленькой }
279     for i := 2 to (Length (A) - 1) do
280         if (A [i] = 'e')
281             and
282             not (Letter (A [i + 1]))
283             and
284             (Letter (A [i - 1]))
285             then
286                 Delete (A, i, 1);

287     { e последняя в слове и в тексте }
288     if (A [Length (A)] = 'e')
289         and
290         (Length (A) > 1)
291         and
292         (Letter (A [Length (A) - 1]))
293     then
294         Delete (A, Length (A), 1);
295 end;

296 Begin
297     Assign (Input, 'input.txt');
298     Assign (Output, 'output.txt');

299     Reset (Input);
300     Read (A);
301     Close (Input);

302 {
    Чтобы избежать многочисленных ошибок, следует в первую очередь
    удалить из текста артикли. Действительно, достаточно сложно
    будет разработать алгоритм, который будет работать в той же
    последовательности, что и реформы: был артикль the, после
    реформы стал th, но его надо удалить.
}

```

Чтобы избежать ошибок Runtime Error при обращении к несуществующему элементу, мы добавим в начало и конец строки по одному пробелу, что гарантирует корректную работу процедуры DelArticle.

```
}  
  
313   A := ' ' + A + ' '  
314   DelArticle (A);  
  
315   { удаляем добавленные пробелы }  
316   while (A [1] = ' '  
317   and  
318   (Length (A) > 1)  
319   do  
320     Delete (A, 1, 1);  
  
321   while (A [Length (A)] = ' '  
322     and  
323     (Length (A) > 0)  
324   do  
325     Delete (A, Length (A), 1);  
  
326   {  
      после удаления артиклей применим реформы первого года –  
      избавимся от буквы с согласно правилам, описанным в условии  
      задачи  
    }  
  
331   FYear (A);  
  
332   { удаляем все удвоенные буквы }  
333   DelDouble (A);  
  
334   { удаляем буквы е в конце слов }  
335   DeIE (A);  
  
336   ReWrite (Output);  
  
337   Write (A);  
  
338   Close (Output);  
339   End.
```

Listing данной задачи опубликован в сети Internet по адресу
<http://www.Best-Listing.ru/color-10-task-648.html>

Sergey Mitrofanov, 11.09.14, 12:09

E-mail: infostar@mail.ru

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