

1

{

File: moulds.dpr

Compiler. Borland Delphi 7.0

Задача. Шаблоном размера n назовем строку длины n , каждый из символов которой входит в множество $[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f, g, ?]$

Шаблоны преобразуются в строки из цифр по следующим правилам:

- символы от 0 до 9 могут быть преобразованы только сами в себя;
- символ a может преобразован в любой из символов 0, 1, 2, 3;
- символ b может преобразован в любой из символов 1, 2, 3, 4;
- символ c может преобразован в любой из символов 2, 3, 4, 5;
- символ d может преобразован в любой из символов 3, 4, 5, 6;
- символ e может преобразован в любой из символов 4, 5, 6, 7;
- символ f может преобразован в любой из символов 5, 6, 7, 8;
- символ g может преобразован в любой из символов 6, 7, 8, 9;
- символ $?$ может преобразован в любой из символов от 0 до 9;

Даны два шаблона: r_1 и r_2 . Рассмотрим множество S_1 строк, которые могут быть получены из r_1 по описанным правилам, и множество S_2 строк, которые могут быть получены из r_2 . Необходимо найти количество строк, входящих в оба этих множества.

Первая строка входного файла Input.txt содержит шаблон r_1 , вторая – шаблон r_2 . Шаблоны имеют одинаковый положительный размер, не больше 9.

В выходной файл Output.txt выведите ответ на задачу.

Input.txt	Output.txt
abc	0
999	

Решение. Калмыков Вадим (ProCrypt),
г. Сургут, ЦНИТ "Северная Звезда",
17:10, 09.06.2008

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

Editor. MSP, 15.12.09, 11:29; LIST 6.0, (c) 1999–2009
}

```

48 Program Moulds;
49 Var
50     p1,
51     p2 : String [9]; // Шаблоны
52     N : Integer; // количество строк, входящих в оба множества
53     i : byte; // счетчик циклов
54     // функция записывающая в строку все возможные преобразования
55     // символа
56 Function Mould (c : Char) : String;
57 begin
58     // цифры могут быть преобразованы только сами в себя
59     if (c >= '0')
60         and
61         (c <= '9')
62     then
63         Mould := c;
64     if (c = 'a')
65     then
66         Mould := '0123';
67     if (c = 'b')
68     then
69         Mould := '1234';
70     if (c = 'c')
71     then
72         Mould := '2345';
73     if (c = 'd')

```

```

74     then
75         Mould := '3456';

76     if (c = 'e')
77     then
78         Mould := '4567';

79     if (c = 'f')
80     then
81         Mould := '5678';

82     if (c = 'g')
83     then
84         Mould := '6789';

85     if (c = '?')
86     then
87         Mould := '0123456789';
88 end;

89 // функция определяющая пересечение множеств. Cross возвращает число
90 // элементов множества str1 принадлежащих также множеству str2
91 Function Cross (str1, str2 : String) : byte;

92 var
93     i,          // счетчик цикла
94     n : byte; // число общих элементов

95 begin
96     n := 0;
97     for i := 1 to Length (str1) do
98         if (Pos (str1 [i], str2) <> 0)
99         then
100             Inc (n);

101     Cross := n;
102 end;

103 Begin
104     Assign (Input, 'input.txt');
105     Assign (Output, 'output.txt');

106     Reset (Input);
107     ReadLn (p1);
108     ReadLn (p2);
109     Close (Input);

110     {
        Если строка S входит в оба множества, то p1 [i] и p [2] можно

```

преобразовать в $S [i]$. Количество общих символов, полученных из $p1 [i]$ и $p2 [i]$ определяется количеством общих возможных преобразований, то есть равна числу общих элементов множеств, значения которых могут принимать $p1 [i]$ и $p2 [i]$. Аналогично определяется и число общих элементов $p1 [i + 1]$ и $p2 [i + 1]$. Пусть $p1 [i]$ и $p2 [i]$ имеют n общих элементов множеств, а $p1 [i + 1]$ и $p2 [i + 1]$ – m . $p1 [i]$ пробегает n значений, в это время следующий за ним символ $p1 [i + 1]$ пробегает m значений. Тогда общее количество вариантов $m * n$ (m вариантов на каждом значении во время пробега n , число пробегов – n раз). Аналогично k раз повторится $p1 [i + 2]$ элемент, тогда уже будет $n * m * k$.

 Таким образом, количество возможных строк, принадлежащих обоим множествам, определяется формулой $N = A1 * A2 * A3 . . . * An$, где – Ai – число общих элементов $p1 [i]$ и $p2 [i]$.

```

}

130 // считаем количество общих строк, входящих в оба множества
131 N := 1; // начальный множитель
132 for i := 1 to Length (p1) do
133   N := N * Cross (Mould (p1 [i]), Mould (p2 [i]));

134   Rewrite (Output);
135   Write (N);
136
137   Close (Output);
138 End.
```

Listing данной задачи опубликован в сети Internet по адресу
<http://www.Best-Listing.ru/color-10-task-673.html>

Sergey Mitrofanov, 19.09.14, 21:05

E-mail: infostar@mail.ru

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