

Тема: Строки

Вариант: 2.1.2

Задача: Написать программу, которая находит все слова в тексте, подходящие под заданный шаблон.

В шаблоне могут быть использованы следующие спецсимволы:

1. “\d” соответствует любому цифровому символу.
2. “\D” соответствует символу буквы английского алфавита.
3. Любой конкретный символ, обозначающий вхождение самого этого символа: ‘а’, ‘b’, ‘c’, и т.д.;
В качестве конкретного символа не используются ‘\’, ‘~’, ‘[’, ‘]’, ‘(’, ‘)’, ‘*’.

Кроме того, в шаблоне могут быть использованы следующие конструкции:

1. ~x, где x – это конкретный символ. Обозначает, что на данной позиции может стоять любой символ кроме x (снова запрещены символы ‘\’, ‘~’, ‘[’, ‘]’, ‘(’, ‘)’, ‘*’).
2. [N*(...)], где N – натуральное число, а в круглых скобках записано выражение по общим правилам. Означает, что вместо данного выражения в квадратных скобках должно стоять N штук выражений в круглых скобках.

Вложенные конструкции [] и ~ в шаблонах не используются

Пример:

Под шаблон “a[3*(\d\D)]~b” подходят строки: “a1z3q4P9”, “a1p1p1pa”, “a9q8p7o1” и т.д.

Формат входных данных:

В первой строке входного файла записан шаблон по приведенным выше правилам.

Считать, что шаблон ограничен 64 символами.

Далее записано число K.

В конце файла идет K строк. Считать, что каждая строка ограничена 1024 символами.

Формат выходных данных:

В выходной файл через пробел записать номера строк, подходящих под указанный шаблон.

Нумерация строк идет с нуля. Указывать строки в том же порядке, в каком они были записаны во входных данных. Если никакая из строк не подошла, вывести “none”.

Пример входных и выходных данных:

input.txt	output.txt
a[3*(\d\D)]~b 5 a33b a1z3q4P9 b1z3q4Pa a9q8p7o1 a1A2B3Ca	1 3 4
123\Dabc 3 123abc 123Qabc 1239abc	1
[10*(\d)] 3 12345 Abc 12c	none

Дополнительное задание:

Добавить возможность использования конструкции <...>* допускающей произвольное количество вхождений выражения в треугольных скобках подряд.