

1. Пусть переменная `var` имеет значение `abc123`. Определите, что возвращают следующие выражения - `true` или `false`:

`\$var =~ /. /` : true.

Метасимвол `.` совпадает с любым символом, кроме символа перевода строки.

`\$var =~ /[A-Z]*/` : true.

Конструкция `[]` определяет список возможных символов для позиции, а метасимвол `*` означает любое количество повторений стоящего до него символа, в том числе и нулевое; таким образом, данное регулярное выражение соответствует всему.

`\$var =~ /(\d)2(\1)/` : false.

Конструкция `()` определяет элементы, которые следует сохранить для использования в обратных ссылках, доступных по символу `\n`, где `n` – целое число от единицы.

`\$var =~ /abc\$/` : false.

Метасимвол `$` является ‘якорем’, знаменующим конец строки; данное выражение проверяет окончание строки.

`\$var =~ /1234?/` : true.

Метасимвол `\?` определяет для символа, стоящего в позиции перед метасимволом, возможность нулевого или единичного повторения.

2. Пусть `\$var` имеет значение `abc123abc`. Какое значение примет `\$var` после следующих подстановок?

`\$var =~ s/abc/def/`; : `def123abc`.

Оператор подстановки без модификаторов изменяет в исходной строке первое вхождение шаблона.

`\$var =~ s/[a-z]+/X/g`; : `XXX123XXX`.

С модификатором `g` (global) замена применяется к каждому вхождению.

`\$var =~ s/B/W/i`; : `aWc123abc`.

С модификатором `i` игнорируется регистр.

`\$var =~ s/(.)\d.*\1/d/`; : `abd`.

Символ `\d` эквивалентен диапазону цифр `[0-9]`; выражение в круглых скобках запоминается для работы обратной ссылки `\1`. Таким образом, производится замена строки от символа перед цифрой до этого же символа через произвольное количество позиций на символ `d`.

`\$var =~ s/(\d+)/$1*2/e`; : `abc246abc`.

Модификатор `e` (eval) изменяет поведение оператора таким образом, что выражение-замена выполняется в качестве Perl-скрипта.

3. Чему соответствуют следующие шаблоны:

`/a|bc*/` : `aaaaaaccccc`, `asdbcccasd`, `bc`.

`/[\d]{1,3}/` : `1`, `834:32134`.

`/\bc[aou]t\b/` : `cat`, `:cut:`.

`/(xy+z)\.\1/` : `xyyyyyyyz.xyyyyyyyz`, `43352xyyyyyyyz.xyyyyyyzewre`

`/~\$/` : `$ getent passwd`, `$$$$`.

4. Написать шаблоны, соответствующие следующим примерам:

4.1. Не менее 5 маленьких латинских букв: `m/[a-z]{5,}/`.

4.2. Цифра 1 или слово "one" (в любом регистре): `m/1|(one)/i`.

4.3. Число, возможно, дробное (с десятичной точкой): `m/\s+|^[-+]?(\d+(\.\d+)?)?\s+/$`.

4.4. Любая буква, за которой следует гласная, повторяется еще раз: `m/([a-z])[aeiouy]\1/`.

4.5. Хотя бы один "+": `m/\++/`