

Практикум №1

Регулярные выражения

В каждой задаче нужно реализовать на языке C++ или Python некоторый алгоритм обработки регулярных выражений. В каждой задаче аргументами являются строка в алфавите $\{a, b, c, 1, ., +, *\}$, а также некоторые дополнительные параметры. Если задача предполагает ответ “да/нет”, то необходимо вывести YES в случае положительного ответа и NO — в случае отрицательного. В случае, если ответ является целым числом или словом, необходимо вывести это число или слово. В случае, если таких числа или слова не существует, необходимо вывести INF. В случае, если входная строка не является корректным регулярным выражением в обратной польской записи, необходимо выдать сообщение ERROR об ошибке. Дополнительные случаи оговорены непосредственно при формулировке задачи.

В дальнейшем предполагается, что первым компонентом входа является регулярное выражение α в обратной польской записи, задающее язык L . Условия задач:

1. Даны α и натуральные числа k, l , такие что $0 \leq l < k$. Проверить, содержит ли язык L слова, чья длина равна l по модулю k .
2. Даны α и натуральные числа k, l , такие что $0 \leq l < k$. Вывести минимальное число n , равное l по модулю k , такое что L содержит слова длины n .
3. Даны α , буква x и натуральное число k . Вывести, есть ли в языке L слова, содержащие префикс x^k .
4. Даны α , буква x и натуральное число k . Вывести, есть ли в языке L слова, содержащие подслово x^k .
5. Даны α , буква x и натуральное число k . Вывести, есть ли в языке L слова, содержащие суффикс x^k .
6. Даны α , буква x и натуральное число k . Вывести, есть ли в языке L слова, содержащие ровно k букв x .
7. Даны α , буква x и натуральное число k . Вывести, есть ли в языке L слова, содержащие кратное k число букв x .
8. Даны α , буква x и натуральное число k . Вывести длину кратчайшего слова из языка L , содержащего префикс x^k .
9. Даны α , буква x и натуральное число k . Вывести длину кратчайшего слова из языка L , содержащего подслово x^k .
10. Даны α , буква x и натуральное число k . Вывести длину кратчайшего слова из языка L , содержащего суффикс x^k .
11. Даны α , буква x и натуральное число k . Вывести длину кратчайшего слова из языка L , содержащего ровно k букв x .
12. Даны α и слово $u \in \{a, b, c\}^*$. Найти длину самого длинного префикса u , принадлежащего L .
13. Даны α и слово $u \in \{a, b, c\}^*$. Найти длину самого длинного подслова u , принадлежащего L .
14. Даны α и слово $u \in \{a, b, c\}^*$. Найти длину самого длинного суффикса u , принадлежащего L .

15. Даны α и слово $u \in \{a, b, c\}^*$. Найти длину самого длинного префикса u , являющегося также префиксом некоторого слова в L .
16. Даны α и слово $u \in \{a, b, c\}^*$. Найти длину самого длинного подслова u , являющегося также подсловом некоторого слова в L .
17. Даны α и слово $u \in \{a, b, c\}^*$. Найти длину самого длинного суффикса u , являющегося также суффиксом некоторого слова в L .
18. Даны α и буква x . Найти максимальное k , такое что в L есть слова, начинающиеся с x^k .
19. Даны α и буква x . Найти максимальное k , такое что в L есть слова, заканчивающиеся на x^k .
20. Даны α и буква x . Найти максимальное k , такое что в L есть слова, содержащие подслово x^k .

Входные данные получаются из `stdin` и выводятся в `stdout`. Программа должна компилироваться под `gnu g++ 9.3.0` или `clang++ 10.0.0` в случае языка C++(C) или интерпретироваться стандартным интерпретатором Python 3.8.5. Код должен удовлетворять разумным требованиям к стилю написания и вычислительной эффективности. Использование библиотек для работы с регулярными выражениями запрещено!

Код программы должен быть размещён в виде приватного репозитория на GitHub с открытым доступом на редактирование для преподавателя. Желательно, чтобы репозиторий назывался `mipt-flat-2020-practice1`. Помимо файлов с решением задачи, в репозитории должен присутствовать README-файл в формате `.md` с описанием идеи алгоритма и оценкой сложности его работы. Если программа использует `makefile` или его аналоги, команда, которой производится сборка, должна также быть указана в README.

Сдача будет проводиться очно в два этапа — сначала будет производиться изначальная «защита» кода с доказательством его корректности и сложности работы, после чего будут выданы комментарии по самому коду (как по организации кода в целом, так и по кодстайлу). Чем меньше будет комментариев, тем меньше итераций будет дальше!

Срок, до которого должна быть предоставлен изначальный код — **06.11.2020, 23:59 МСК**. Финальная защита кода должна быть произведена до **18.11.2020, 23:59**.

Тесты

Номер задачи	Вход	Выход
1	$ab + c.aba. * .bac. + . + * \quad 3 \quad 2$	YES
	$acb..bab.c. * .ab.ba. + . + * a. \quad 3 \quad 0$	NO
2	$ab + c.aba. * .bac. + . + * \quad 3 \quad 1$	4
	$acb..bab.c. * .ab.ba. + . + * a. \quad 3 \quad 0$	INF
3	$ab + c.aba. * .bac. + . + * \quad a \quad 2$	YES
	$acb..bab.c. * .ab.ba. + . + * a. \quad b \quad 3$	NO
4	$ab + c.aba. * .bac. + . + * \quad a \quad 4$	NO
	$acb..bab.c. * .ab.ba. + . + * a. \quad b \quad 2$	YES
5	$ab + c.aba. * .bac. + . + * \quad a \quad 2$	NO
	$acb..bab.c. * .ab.ba. + . + * a. \quad c \quad 0$	YES
6	$ab + c.aba. * .bac. + . + * \quad a \quad 2$	YES
	$acb..bab.c. * .ab.ba. + . + * a. \quad a \quad 0$	NO
7	$ab + c.aba. * .bac. + . + * \quad a \quad 2$	YES
	$aba. * .a. * ab1 + .. \quad a \quad 2$	NO
8	$ab + c.aba. * .bac. + . + * \quad c \quad 4$	INF
	$acb..bab.c. * .ab.ba. + . + * a. \quad b \quad 2$	4
9	$ab + c.aba. * .bac. + . + * \quad b \quad 2$	4
	$acb..bab.c. * .ab.ba. + . + * a. \quad b \quad 3$	7
10	$ab + c.aba. * .bac. + . + * \quad b \quad 2$	INF
	$acb..bab.c. * .ab.ba. + . + * a. \quad a \quad 2$	4
11	$ab + c.aba. * .bac. + . + * \quad b \quad 2$	4
	$acb..bab.c. * .ab.ba. + . + * a. \quad b \quad 3$	7
12	$ab + c.aba. * .bac. + . + * \quad abacb$	4
	$acb..bab.c. * .ab.ba. + . + * a. \quad cb$	0
13	$ab + c.aba. * .bac. + . + * \quad babc$	2
	$acb..bab.c. * .ab.ba. + . + * a. \quad abbaa$	4
14	$ab + c.aba. * .bac. + . + * \quad babc$	2
	$acb..bab.c. * .ab.ba. + . + * a. \quad cbaa$	1
15	$ab + c.aba. * .bac. + . + * \quad abacb$	4
	$acb..bab.c. * .ab.ba. + . + * a. \quad acbac$	4
16	$ab + c.aba. * .bac. + . + * \quad babc$	3
	$acb..bab.c. * .ab.ba. + . + * a. \quad abbaa$	5
17	$ab + c.aba. * .bac. + . + * \quad babc$	2
	$acb..bab.c. * .ab.ba. + . + * a. \quad cbaa$	4
18	$ab + c.aba. * .bac. + . + * \quad a$	2
	$acb..bab.c. * .ab.ba. + . + * a. \quad c$	0
19	$ab + c.aba. * .bac. + . + * \quad b$	1
	$acb..bab.c. * .ab.ba. + . + * a. \quad b$	0
20	$ab + c.aba. * .bac. + . + * \quad a$	2
	$acb..bab.c. * .ab.ba. + . + * a. \quad a$	2