

Чемпионат по программированию: Бэкенд-разработка - Квалификация

🕒 20 окт 2019, 18:05:17

старт: 20 окт 2019, 12:36:19

финиш: 20 окт 2019, 17:36:19

длительность: 05:00:00

начало: 14 окт 2019, 12:00:00

конец: 20 окт 2019, 23:59:59

D. Марсианские промокоды

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

После запуска Яндекс.Такси на Марсе прошло полгода. К марсианскому Новому Году Яндекс.Такси решило подарить марсианам промокоды. Но оказалось, что марсиане не могут использовать земные промокоды, так как сервера на Марсе не работают по земным правилам. Поэтому Яндекс.Такси придумало специальные марсианские промокоды. Марсианский промокод генерируется следующим образом:

1. Генерируется ключ шифрования.
2. Ключ шифрования разбивается на подстроки случайной длины.
3. Из всех подстрок выбираются подстроки длины L ($L > 1$).
4. Выбранные подстроки перемешиваются и конкатенируются.
5. Результат конкатенации является промокодом.

Задача: Необходимо проверить, является ли введенный промокод валидным. Если промокод валиден — нужно вывести «YES». Если невалиден — «NO».

Формат ввода

<длина промокода>

<промокод>

<длина ключа>

<ключ>

<длина подстроки L>

Формат вывода

Валидность промокода: YES или NO

Пример 1

Ввод	Вывод
6	YES
ABCDEA	
6	
EAABCD	
2	

Пример 2

Ввод	Вывод
12	YES
MARS1234MARS	
24	
ASDGRV12MARSS1234VRCMARS	
4	

Пример 3

Ввод	Вывод
12	NO
ABC123123ABC	
9	
ABC123123	
3	

Примечания

Длина $L > 1$.

Алфавит промокода [A-Z0-9].

Длина промокода находится в диапазоне [6, 30].

Длина ключа находится в диапазоне [6, 30].

Длина подстроки $L <$ длина ключа.

Длина промокода кратна L .

Язык Python 3.7.3

Набрать здесь

Отправить файл

```

1 def main():
2     with open('input.txt', 'r') as file:
3         promo_len = int(file.readline())
4         promo = file.readline()
5         key_len = int(file.readline())
6         keys = [file.readline()]
7         L = int(file.readline())
8
9         parts = []
10        for i in range((promo_len // L)):
11            parts.append(promo[L * i:L * (i + 1)])
12
13        count = 0
14        for part in parts:
15            for key in keys:
16                if part in key:
17                    pos_key = key.find(part)
18                    key = key[pos_key:]
19                    keys.append(key[pos_key + L:])
20                    count += 1
21        if count == len(parts):
22            return "YES"
23        else:
24            return "NO"
25
26
27 if __name__ == '__main__':
28     print(main())
29

```

Отправить

Предыдущая