Repeated String ☆

Lilah has a string, s, of lowercase English letters that she repeated infinitely many times.

Given an integer, n, find and print the number of letter a's in the first n letters of Lilah's infinite string.

For example, if the string $s =$ 'abcac' and $n = 10$, the substring we consider is $abcacabcac$, the first 10 characters of her infinite string. There are 4 occurrences of a in the substring.

Function Description

Complete the repeatedString function in the editor below. It should return an integer representing the number of occurrences of a in the prefix of length n in the infinitely repeating string.

repeatedString has the following parameter(s):

- s: a string to repeat
- n: the number of characters to consider

Input Format

The first line contains a single string, s.

The second line contains an integer, n.

Constraints

- $1 \leq |s| \leq 100$
- $1 \leq n \leq 10^{12}$
- For 25% of the test cases, $n \leq 10^6$.

Output Format

Print a single integer denoting the number of letter a's in the first n letters of the infinite string created by repeating s infinitely many times.

Sample Input 0

```
aba
10
```

Sample Output 0

```
7
```

Explanation 0

The first $n = 10$ letters of the infinite string are abaabaabaa. Because there are 7 a's, we print 7 on a new line.

Sample Input 1

```
a
1000000000000
```

Sample Output 1

```
1000000000000
```

Explanation 1

Because all of the first $n = 1000000000000$ letters of the infinite string are a, we print 1000000000000 on a new line.

Author tunyash
Difficulty Easy
Max Score 20
Submitted By 346847

NEED HELP?

View discussions
View editorial
View top submissions

RATE THIS CHALLENGE

☆ ☆ ☆ ☆ ☆

MORE DETAILS

Download problem statement
Download sample test cases
Suggest Edits

Change Theme Python 3

```
#!/bin/python3

import math
import os
import random
import re
import sys

# Complete the repeatedString function below.
def repeatedString(s, n):
    #No. of whole repeatations
    quotient = n//len(s)
    # Extra part (part of tapered repeatation)
    rem = n % len(s)
    #print(quotient)
    #print(rem)
    if n < len(s):
        res = s[0:n+1].count("a")
    else:
        extra = s[0:rem].count("a")
        res = s.count("a")*quotient + extra
    return(res)
```

Line: 37 Col: 1

Upload Code as File Test against custom input Run Code Submit Code