Practice Compete Jobs Rank Leaderboard

Dashboard > Python > Strings > Merge the Tools!

Points: 775 Rank: 8720

Merge the Tools!

byanuj_95

Problem Submissions Leaderboard Discussions Editorial

Consider the following:

- A string, s, of length n where $s = c_0c_1 \dots c_{n-1}$.
- An integer, k, where k is a factor of n.

We can split s into $\frac{n}{k}$ subsegments where each subsegment, t_i, consists of a contiguous block of k characters in s. Then, use each t_i to create string u_i such that:

- The characters in u_i are a subsequence of the characters in t_i.
- Any repeat occurrence of a character is removed from the string such that each character in u_i occurs exactly once. In other words, if the character at some index j in t_i occurs at a previous index $< j$ in t_i, then do not include the character in string u_i.

Given s and k, print $\frac{n}{k}$ lines where each line i denotes string u_i.

Input Format

The first line contains a single string denoting s.
The second line contains an integer, k, denoting the length of each subsegment.

Constraints

- $1 \leq n \leq 10^4$, where n is the length of s
- $1 \leq k \leq n$
- It is guaranteed that n is a multiple of k.

Output Format

Print $\frac{n}{k}$ lines where each line i contains string u_i.

Sample Input

```
AABCAAADA
3
```

Sample Output

```
AB
CA
AD
```

Explanation

String s is split into $\frac{n}{k} = \frac{9}{3} = 3$ equal parts of length $k = 3$. We convert each t_i to u_i by removing any subsequent occurrences non-distinct characters in t_i:

1. $t_0 = \texttt{"AAB"} \rightarrow u_0 = \texttt{"AB"}$
2. $t_1 = \texttt{"CAA"} \rightarrow u_1 = \texttt{"CA"}$
3. $t_2 = \texttt{"ADA"} \rightarrow u_2 = \texttt{"AD"}$

We then print each u_i on a new line.

Medium

Submitted 13560 times
Max Score **40**

Need Help?

- View Discussions
- View Editorial Solution
- View Top Submissions

RATE THIS CHALLENGE

☆ ☆ ☆ ☆ ☆

Download problem statement

Download sample test cases

Suggest Edits

Current Buffer (saved locally, editable)

Python 3

```python
def merge_the_tools(string, k):
    # your code goes here
    str_len = len(string)

    start = 0
    end = k

    while end < str_len + 1:
        try:
            l1 = []
            [l1.append(x) for x in string[start:end] if x not in l1]

            print(''.join(l1))
            start = end
            end += k
        except IndexError:
            print(''.join(l1))
            break

if __name__ == '__main__':↔
```

Line: 7 Col: 1

Upload Code as File

□ Test against custom input

Run Code

Submit Code

Congrats, you solved this challenge!

Challenge your friends:

✓ Test Case #0

✓ Test Case #1

✓ Test Case #2

✓ Test Case #3

✓ Test Case #4

✓ Test Case #5

✓ Test Case #6

✓ Test Case #7

✓ Test Case #8

✓ Test Case #9

✓ Test Case #10

✓ Test Case #11

✓ Test Case #12

✓ Test Case #13

✓ Test Case #14

✓ Test Case #15

✓ Test Case #16

You've earned 40.00 points.

Next Challenge

Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature