

# Prefix Compression ★

Points: 553.1800000000001 Rank: 1150

Problem

Submissions

Leaderboard

Editorial

## RATE THIS CHALLENGE



You are in charge of data transfer between two Data Centers. Each set of data is represented by a pair of strings. Over a period of time you have observed a trend: most of the times both strings share some prefix. You want to utilize this observation to design a data compression algorithm which will be used to reduce amount of data to be transferred.

You are given two strings,  $x$  and  $y$ , representing the data, you need to find the longest common prefix ( $p$ ) of the two strings. Then you will send substring  $p$ ,  $x'$  and  $y'$ , where  $x'$  and  $y'$  are the substring left after stripping  $p$  from them.

For example, if  $x = \text{"abcdefpr"}$  and  $y = \text{"abcpqr"}$ , then  $p = \text{"abc"}$ ,  $x' = \text{"defpr"}$ ,  $y' = \text{"pqr"}$ .

### Input Format

The first line contains a single string denoting  $x$ .

The second line contains a single string denoting  $y$ .

### Constraints

- $x$  and  $y$  will contain only lowercase Latin characters ('a'-'z').
- $1 \leq \text{length}(x), \text{length}(y) \leq 10^5$

### Output Format

In first line, print the length of substring  $p$ , followed by prefix  $p$ . In second line, print the length of substring  $x'$ , followed by substring  $x'$ . Similarly in third line, print the length of substring  $y'$ , followed by substring  $y'$ .

### Sample Input 0

```
abcdefpr
abcpqr
```

### Sample Output 0

```
3 abc
5 defpr
3 pqr
```

### Sample Input 1

```
kitkat
kit
```

### Sample Output 1

```
3 kit
3 kat
0
```

### Sample Input 2



puppy  
puppy

### Sample Output 2

5 puppy  
0  
0

### Explanation

Sample Case 0:

Already explained above in the problem statement.

Sample Case 1:

$p = \text{"kit"}$ , which is also  $y$ . So  $x'$  will be "kat" and  $y'$  will be an empty string.

Sample Case 2:

Because both strings are the same, the prefix will cover both the strings. Thus,  $x'$  and  $y'$  will be empty strings.

[Change Theme](#)

Language

Haskell



```

1  import Control.Monad
2
3  prefixCount :: String -> String -> Int
4  prefixCount [] _ = 0
5  prefixCount _ [] = 0
6  prefixCount (a:as) (b:bs) = if a == b then 1 + prefixCount as bs else 0
7
8  main = do
9      str1 <- getLine
10     str2 <- getLine
11     let i = prefixCount str1 str2
12     putStr $ show i
13     putStrLn $ " " ++ take i str1
14     putStr $ show (length str1 - i)
15     when (length str1 - i > 0) $ putStr $ " " ++ drop i str1
16     putStrLn ""
17     putStr $ show (length str2 - i)
18     when (length str2 - i > 0) $ putStr $ " " ++ drop i str2
19     putStrLn ""
20
21

```

Line: 21 Col: 1

☒ Upload Code as File

☐ Test against custom input

Run Code

Submit Code



[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)

