

BURROWS-WHEELER TRANSFORM

CHALLENGE DESCRIPTION:

You are trying to unpack a text compressed using run-length encoding (the algorithm described in the Compressed Sequence challenge). You are doing everything right, but get absolutely ridiculous results. As it came out, before the compression, somebody used the Burrows-Wheeler algorithm. It allows increasing the number of repeated characters in the text, and thus, increase the efficiency of compression. This method is used by the bzip2 utility.

Let’s examine the line processing using the Burrows-Wheeler algorithm. First of all, we should mark the end of the line with a special character (EOF), it is \$ in this case. Then, generate all possible rotations of the line, sort them alphabetically and construct a new string — Burrows-Wheeler transform — from the last characters of each rotation.

For example, the line ‘easy-peasy’ is transformed as follows:

Input	All rotations	Sorted rotations	Output
easy-peasy\$	easy-peasy\$ \$easy-peasy y\$easy-peas sy\$easy-pea asy\$easy-pe easy\$easy-p peasy\$easy- -peasy\$easy y-peasy\$eas sy-peasy\$ea asy-peasy\$e	\$easy-peasy -peasy\$easy asy\$easy-pe asy-peasy\$e easy\$easy-p easy-peasy\$ peasy\$easy- sy\$easy-pea sy-peasy\$ea y\$easy-peas y-peasy\$eas	yyeep\$-aass

The Burrows-Wheeler algorithm is valuable not only because it allows introducing more repetitions in the line (we can achieve even better results by simple sorting of all characters), but, with Burrows-Wheeler transform you can easily and unambiguously recreate the original text. Your task is to print out the original text.

INPUT SAMPLE:

Your program should accept a path to a file as its first argument. Each line of the file is a text transformed using the Burrows-Wheeler algorithm. The end of each line is marked with ‘|’ symbol and it should be stripped before you start to process the string.

```
oooooooo$  ffffffff      ffffffffuuuuuuuuaaaaaaaaallllllllbbBbbBBb|
edarddddddddddntensr$  ehhhhhhhhhhJ aeaaaaaaaaaalhtf thmbfe          tcwohiahoJ eeec t e |
ooooio,io$Nnssshhho ee o nnkkkkkkii |
```

OUTPUT SAMPLE:

For each test case, print to stdout the original text.

```
Buffalo buffalo Buffalo buffalo buffalo buffalo Buffalo buffalo$
James while John had had had had had had had had had had a better effect on the teacher$
Neko no ko koneko, shishi no ko kojishi$
```

CONSTRAINTS:

1. The number of test cases is 20.
2. The length of each line is from 5 to 1500 characters.
3. The sorting order of the characters is defined by their ASCII code, that is the character \$ (code 36) goes before the letters of the alphabet.