Detect rotated strings

Read two strings from `stdin`.
Output `Yes` if one string is a rotated version of the other.
Otherwise output `No`

Testcases

Input

```
CodeGolf GolfCode
```

Output

```
Yes
```

Input

```
stackexchange changestackex
```

Output

```
Yes
```

Input

```
stackexchange changestack
```

Output

```
No
```

Input

```
Hello World
```

Output

```
No
```

code-golf string

asked Mar 9 '11 at 11:53

gnibbler
12.4k 2 34 71

So (*abcdefAB*, *ABabcdef*) is a "YES"? – Eelvex Mar 9 '11 at 12:06

Should it really be a rotation or is a combination fine too? eg. what will `Stackexchange Stackchangeex` return? – jpjacobs Mar 9 '11 at 12:08

@Eelvex, yes. @jpjacobs, It would return `No`. The rotation is a shift, like those LED scrolling signs – gnibbler Mar 9 '11 at 12:19

Are the strings always whitespace-free and separated by whitespace? – Joey Mar 9 '11 at 15:20

More specifically, what characters are allowed in those strings? – Joey Mar 9 '11 at 15:48

36 Answers

1 2 next

APL (28)

Takes input on two lines.

```
'No' 'Yes'[1+(⊂⍞)∊⌽∘A¨⍳⍴A←⍞]
```

Explanation:

- `A←⍞` : read a line of input and store it in A
- `⌽∘A¨⍳⍴A` : Rotate A by x, for each x in [1..length A]. Gives a list, i.e. `estT stTe tTes Test`
- `(⊂⍞)∊` : read another line of input, and see if it is in this list.
- `1+` : add one to this, giving 1 if the strings were not rotated and 2 if they were
- `'No' 'Yes'[ ... ]` : Select either the first or second element from the list `'No' 'Yes'` depending on whether the strings were rotated or not.
- This value is output automatically.

answered May 29 '12 at 11:53
marinus
27.5k 6 51 95

Ruby ~~49~~ 41

```
a,b=$*;puts (a*2).sub(b,'')==a ?:yes: :no
```

Edit: replaced gets.split by $*

edited Mar 15 '11 at 8:55

answered Mar 9 '11 at 21:31
steenslag
1,627 8 18

That's an ingenious idea. – Joey Mar 9 '11 at 21:40

Very clever. :) – st0le Mar 18 '11 at 4:56

Python-70 bytes

```
a,b=raw_input().split()
print ['No','Yes'][a in b*2and len(a)==len(b)]
```

Testing ...

edited Mar 9 '11 at 18:43

answered Mar 9 '11 at 14:45
Quixotic
1,466 7 22

+1 Nice, selecting the result from an array is clever! :-) – Tom Wijsman Mar 9 '11 at 15:24

3 The question states you should read two strings from `stdin` , which this solution does not. – Ventero Mar 9 '11 at 15:36

@Ventero:Fixed. – Quixotic Mar 9 '11 at 18:43

J, 47

```
y=:>2{ARGV
(>1{ARGV e.1|.^:(i.#y)y){'No',:'Yes'
```

edited Mar 9 '11 at 13:27

answered Mar 9 '11 at 13:11
Eelvex
4,005 1 15 36

Why the two J answers? – J B Mar 9 '11 at 19:49

@JB: because this one uses the buildin rotate. Both answers su^H^H are not so good btw. There is a *lot* of room for golfing. – Eelvex Mar 9 '11 at 19:57

why the other one, then, I'm tempted to ask? :-) – J B Mar 9 '11 at 20:06

@JB: because I thought this one is barely legal (:p) [while the other one extends nicely to lisp.] – Eelvex Mar 9 '11 at 20:13

errr... the other one seems to read input from the command-line as well – J B Mar 9 '11 at 20:15

According to the spec (same string lengths):

Perl, ~~42~~ 43 chars

```
$.=pop;$_=(pop)x2;print+(qw'yes no')[!/$./]
```

If different sized strings are allowed, the solution would be:

Perl, 47 chars

```
$.=(pop)x8;$_=(pop)x9;print+(qw'yes no')[!/$./]
```

rbo

edited Mar 9 '11 at 15:33

answered Mar 9 '11 at 14:11

rubber boots
994 6 6

Same problem as the GolfScript solution: If you input `nn nfn`, you get `Yes`, which is wrong. – Tom Wijsman Mar 9 '11 at 14:17

1 seems to be ok (I missed the'!' in the first version) "nn nfn" => **no** "CodeGolf GolfCode" => **yes** – rubber boots Mar 9 '11 at 14:22

Python 70 Characters

```
a,b=raw_input().split()
print'YNeos'[len(a)<>len(b)or a not in 2*b::2]
```

Thanks to gnibbler for the slice trick.

edited Mar 13 '11 at 14:26

answered Mar 9 '11 at 14:11

fR0DDY
3,133 12 29

1 Same problem as the GolfScript solution: If you input `nn nfn`, you get `Yes`, which is wrong. – Tom Wijsman Mar 9 '11 at 14:28

@TomWij Thanks for finding the bug. Corrected. Should work now. – fR0DDY Mar 9 '11 at 15:14

You can replace `<>` by `-` as that will also result in `0` if they are of equal length. – Tom Wijsman Mar 9 '11 at 15:28

But what if they are not of equal length? Then it doesn't work so well :-) – hallvabo Mar 13 '11 at 13:38

@hallvabo then the strings are not rotated version of each other. – fR0DDY Mar 13 '11 at 13:52

Golfscript, 31

```
' '/:)~,\,=)~.+\/,(&'Yes''No'if
```

This one check length first, so it should work as expected.

edited Mar 25 '11 at 1:32

answered Mar 19 '11 at 6:23

YOU
3,592 2 12 21

`:)` and `=)` +1 for very happy code – TùxCräftîñg Oct 30 '16 at 23:48

J, 57

```
{&('No';'Yes')@-:/@:((/:~@(|."0 _~i.&$))&.>)&.(;:&stdin)_
```

Sample use:

```
$ echo -n CodeGolf GolfCode | jconsole rotate.ijs
Yes
$ echo -n stackexchange changestackex | jconsole rotate.ijs
Yes
$ echo -n stackexchange changestack | jconsole rotate.ijs
No
$ echo -n Hello World | jconsole rotate.ijs
No
```

answered Mar 9 '11 at 20:35

J B
8,966 17 50

JavaScript, 51

```
function f(a,b)a&&(a+a).replace(b,"")==a?"Yes":"No"
```

JavaScript doesn't have a canonical host, so this answer is written as a function of two arguments. The score goes up to 60 if we disallow JS 1.7 features (expression closures).

In the SpiderMonkey shell this would be (for a score of 71):

```
[a,b]=readline().split(" ");print(a&&(a+a).replace(b,"")==a?"Yes":"No")
```

answered Mar 13 '11 at 22:11
ecatmur
1,495 8 13

5 years later and now you can use the `=>` function notation ;) – J Atkin Feb 23 '16 at 1:28

J, 84

```
y=:(>1{ARGV),:(>2{ARGV)
((0{y)e.(y&((]$0{[),(]-~[:}.[:$[)$1{[)/.i.}.$y)){'No',:'Yes'
```

answered Mar 9 '11 at 12:52
Eelvex
4,005 1 15 36

JavaScript (120 chars)

```
function f(a,b) {for (i=0,A=a.split("");A.join("")!=b&&i++
<a.length;A.push(A.shift()));return A.join("")==b?'Yes':'No';}
```

Output:

```
f('CodeGolf','GolfCode'); //Yes
f('stackexchange','changestackex'); //Yes
f('stackexchange','changestack'); //No
f('Hello','World'); //No
f('nn','nBn'); //No
```

answered Mar 9 '11 at 14:44
Sergio Cinos
121 2

Ruby, 58 (62) characters

```
a,b=gets.split;$><<(a.size==b.size&&/#{a}/=~b*2?:Yes: :No)
```

This solution assumes the input contains only alphanumeric characters (actually everything that doesn't have a special meaning inside a regular expression is ok).

A solution that doesn't have this constraint is 4 characters longer

```
a,b=gets.split;$><<(a.size==b.size&&(b*2).index(a)?:Yes: :No)
```

edited Mar 9 '11 at 15:47

answered Mar 9 '11 at 15:34
Ventero
9,386 2 17 52

Windows PowerShell, 76

```
$a,$b=-split$input
('No','Yes')[+!($a.length-$b.length)*"$b$b".contains($a)]
```

answered Mar 9 '11 at 17:11
Joey
8,903 31 60

Python, 71

```
a,b=raw_input().split()
print'Yes'if a in b*2and len(a)==len(b)else'No'
```

edited Mar 9 '11 at 18:44

answered Mar 9 '11 at 13:22
Tom Wijsman
121 4

Same problem as the GolfScript solution: If you input `nn nfn`, you get `Yes`, which is wrong. – Timwi Mar 9 '11 at 14:08

Problem has been solved, remains low... :-) – Tom Wijsman Mar 9 '11 at 14:17

Doesn't read from stdin as specified. – geoffspear Mar 9 '11 at 16:57

Now it does... :-) – Tom Wijsman Mar 9 '11 at 18:59

PHP, 61

```
<?echo preg_match('/^(.+)(.*) \\2\\1$/',fgets(STDIN))?Yes:No;
```

edited Mar 20 '11 at 16:45

answered Mar 20 '11 at 16:30
Daniel
151 3

Ruby, 41

```
puts gets =~ /^(.+)(.*) \2\1$/ ?:Yes: :No
```

edited Apr 11 '11 at 20:16

answered Mar 20 '11 at 16:40
Daniel
151 3

Doesn't print anything, results in ':No' for input 'aaa aaa' (on my machine). The regexp approach could be a good idea though. – steenslag Mar 25 '11 at 0:43

Fix to actually print and input from stdin instead of args: `puts gets =~ /^(.+)(.*) \2\1$/ ?:Yes: :No` - ups it to 41 chars. – Nemo157 Mar 25 '11 at 4:12

Thanks, I'll make the change @Nemo157 . – Daniel Apr 11 '11 at 20:15

Haskell (~~98~~ 96 chars)

```
g x y@(t:r)(z:w)|x==y="Yes"|1>0=g x(r++[t])w
g _ _[]="No"
f(x:y:_)=g x y y
main=interact$f.words
```

edited Apr 15 '11 at 3:41

answered Mar 15 '11 at 8:16
jloy
631 3 5

Fixed to say Yes/No and improved (91 chars)

```
import List;f[a,b]|a`elem`[x++y|x<-tails b|y<-inits b]="Yes";f
_="No";main=interact$f.words
```

original Haskell (92 chars) that says True/False

```
import Data.List;f(a:b:_)=any(a==)$zipWith(++)(tails b)(inits
b);main=interact(show.f.words)
```

edited Apr 15 '11 at 14:25

answered Mar 30 '11 at 18:59
Chris Kuklewicz
121 2

You can save a few characters with `import List` and also by using `$` instead of parens around the argument to `interact` . But your program outputs `True` and `False` instead of the required `Yes` and `No` , so that might take a few extra chars to fix. – jloy Apr 15 '11 at 3:38

Thanks for the constructive criticism, jloy – Chris Kuklewicz Apr 15 '11 at 14:20

Q (~~50~~ 43 chars)

```
{`No`Yes x in((!)(#)y)rotate\:y}." "vs(0:)0
```

edited May 30 '12 at 15:39

answered May 29 '12 at 21:32
skeevey
3,633 10 21

Scala 78

```
val b=readLine split " "
print(b(0).size==b(1).size&&(b(0)+b(0)contains b(1)))
```

It's a shame about the size check, without it the count drops to **54**

```
val a=readLine split " "
print(a(0)+a(0)contains a(1))
```

answered May 31 '12 at 15:53

Don Mackenzie
131 2

`"""val b=readLine split " " print(b(0).sorted==b(1).sorted)""".length` yields 56 – user unknown Jun 1 '12 at 3:49

Lua 115 chars

```
a,b=io.read():match"(%w+) (%w+)"c=b repeat c=c:sub(2,-1)..c:sub(1,1) s=s or a==c until
b==c print(s and"Yes"or"No")
```

answered Mar 9 '11 at 12:16

jpjacobs
3,362 12 15

C program - 146

```
char b[99],c[99],*p,*q;main(n){q=(p=b+(n=strlen(gets(c))))+n;sprintf(b,"%s%s"
,c,c);for(gets(c);p>b&&strcmp(p,c);--p,*--q=0);puts(p>b?"Yes":"No");}
```

answered Mar 9 '11 at 18:26

Joey Adams
7,628 2 26 51

PHP, 82 characters

```
<?$s=split(" ",fgets(STDIN));echo str_replace($s[1],"",$s[0].$s[0])==$s[0]?Yes:No;
```

answered Mar 13 '11 at 22:33

Kevin Brown
3,278 4 25 58

perl, 123 chars

```
@s1=split(//,shift);
$s2=shift;
$i=0;
while($i<=@s1){
    if(join("",@s1) eq $s2){die "yes";}
    unshift @s1,pop @s1;
    $i++;
}
die "no";
```

answered Mar 18 '11 at 0:27

sogart
117 3

Ruby, ~~30~~ 37

```
gets
puts~/^(.+)(.*) \2\1$/?:Yes: :No
```

A version that prints "true" and "false" instead of "yes" and "no":

```
gets
p !! ~/^(.+)(.*) \2\1$/
```

Both of these work with different-length strings (unlike the old one)

edited Mar 26 '11 at 15:00

answered Mar 24 '11 at 23:25

Lowjacker
3,946 9 18

Does work if the two strings have the same length, but fails on input like 'golfcode golf'. – steenslag Mar 25 '11 at 0:37

Python 2, 86 Characters

```
a,b=raw_input().split()
print"Yes"if any(a==b[n:]+b[:n]for n in range(len(a)))else"No"
```

answered Mar 30 '11 at 9:55

Jeremy Banks ♦
110 5

Perl (just a quick fix)

A fix to rubber boots' solution, being a new user that I am I can't comment yet so I'll just post a new answer.

As the mentioned method uses a regular expression constructed from user input, it is possible to perform a small regex injection, as follows:

```
> perl -le '$.=pop;$_=(pop)x2;print+(qw/yes no/)[!/$./]' anything '.*'
yes
```

The fix is to use \Q (known also as quotemeta):

```
> perl -le '$.=pop;$_=(pop)x2;print+(qw/yes no/)[!/\Q$./]' anything '.*'
no
```

The code itself could be further shortened using 'say' but this is left as an exercise to the reader :)

edited Apr 12 '11 at 22:45

answered Apr 12 '11 at 18:13

chinese perl goth
1,041 6 11

Furthermore, instead of `print+(qw/yes no/)[` you can probably write `print qw(yes no)[` which is two characters shorter. – Timwi Apr 12 '11 at 22:58

Clojure, 95

The function `rotated?` simply tests one string against all possible rotations of the other. Brute force.

```
(defn rotated?
  [a b]
  (let[l (count a)
       b (seq b)]
    (->> a cycle (partition l 1) (take l) (some #{b}))))
```

I personally don't care where the input strings come from, but if it has to be stdin...

```
(apply rotated? (.split (read-line) " "))
```

This would add 38 to character count, if strings were whitespace separated.

```
(rotated? (read-line) (read-line))
```

This would add 32 to character count, if strings were newline separated. The function `rotated?` works with any characters in the strings.

answered Apr 13 '11 at 12:11

Meikel
51 2

Bash, ~~70~~ ~~65~~ ~~59~~ 55 characters

new approach:

```
read a b;c=$a$a;[[ ${c/$b/} == $a ]]&&echo yes||echo no
```

edited May 31 '12 at 23:13

answered May 29 '12 at 21:12

daniero
12.2k 3 28 65

solution 2 fails on `echo ofofo fofof | bash rotated.sh` - doesn't it? Else I would remove even more WS: `read a b[[ $a$a =~ $b&&${#a}-eq${#b} ]]&& echo yes||echo no` . I had a very similar idea, and think it works. – user unknown May 30 '12 at 6:21

You're right, more ws can be removed, and the paratheses! Need a `;` after the read tho. – daniero May 30 '12 at 14:16

@userunknown - hah, got'cha ;) – daniero May 31 '12 at 23:13

You can remove a blank before and one behind `==` . – user unknown Jun 1 '12 at 3:41

1 I tried, but then it always return 'yes'. – daniero Jun 7 '12 at 14:46

bash 56

```
read a b
[[ $a$a =~ $b&&$b$b =~ $a ]]&&echo Yes||echo No
```

edited Jun 7 '12 at 14:55

answered May 30 '12 at 6:22

user unknown
3,797 20 26

1 2 next