

**PRACTICE** 

COMPETE

JOBS

LEADERBOARD

Q Search



o punnammani1 ∨

## Group(), Groups() & Groupdict() ☆

46/115 challenges solved

Rank: 11234 | Points: 765 ①



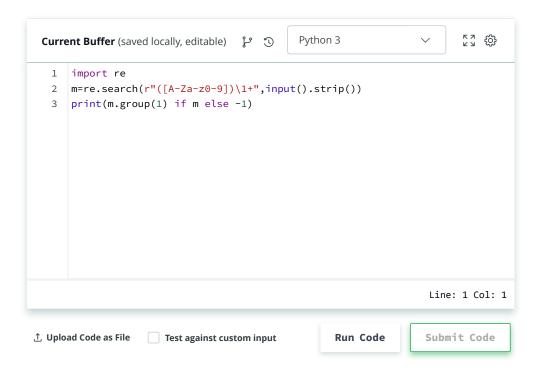
Your Group(), Groups() & Groupdict() submission got 20.00 points. Share

Try the next challenge | Try a Random Challenge

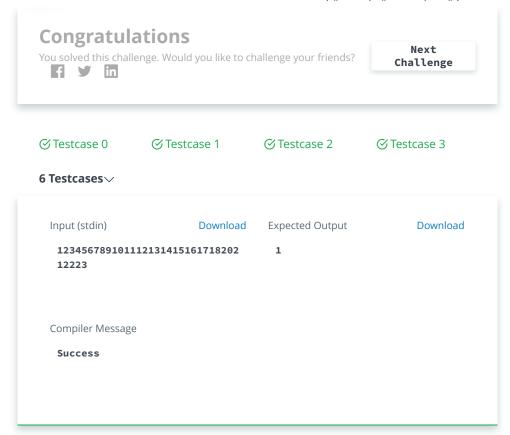
		Leaderboard	Discussions Edito		
group()					
A <i>group()</i> expression	n returns one or more s	ubgroups of the match.			
Code					
>>> import re					
<pre>&gt;&gt;&gt; m = re.match(r'(\w+)@(\w+)\.(\w+)','username@hackerrank.com') &gt;&gt;&gt; m.group(0)  # The entire match</pre>					
<pre>&gt;&gt;&gt; m.group(0 'username@hac</pre>	•	match			
<pre>&gt;&gt;&gt; m.group(1 'username'</pre>		parenthesized subgr	oup.		
>>> m.group(2	) # The second	parenthesized subg	roup.		
>>> m.group(3)  # The third parenthesized subgroup.					
'com' >>> m.group(1	.2.3) # Multiple a	arguments give us a	tuple.		
	<pre>&gt;&gt;&gt; m.group(1,2,3) # Multiple arguments give us a tuple. ('username', 'hackerrank', 'com')</pre>				
-	on returns a tunle conta	ining all the subgroups	of the match		
groups() A groups() expression	on returns a tuple conta	ining all the subgroups	of the match.		
A <i>groups()</i> expression	on returns a tuple conta	ining all the subgroups	of the match.		
A groups() expression  Code  >>> import re	on returns a tuple conta tch(r'(\w+)@(\w+)\.				
A groups() expression  Code  >>> import re >>> m = re.mat >>> m.groups()	tch(r'(\w+)@(\w+)\. )	(\w+)','username@h			
A groups() expression  Code  >>> import re >>> m = re.mat >>> m.groups()	tch(r'(\w+)@(\w+)\.	(\w+)','username@h			
A groups() expression  Code  >>> import re >>> m = re.mat >>> m.groups()	tch(r'(\w+)@(\w+)\. )	(\w+)','username@h			
A groups() expression  Code  >>> import re >>> m = re.man >>> m.groups( ('username',  groupdict()	tch(r'(\w+)@(\w+)\. ) 'hackerrank', 'com'	(\w+)','username@h			
A groups() expression  Code  >>> import re >>> m = re.man >>> m.groups( ('username',  groupdict()	tch(r'(\w+)@(\w+)\. ) 'hackerrank', 'com' ssion returns a dictionar	(\w+)','username@h	ackerrank.com')		
A groups() expression  Code  >>> import re >>> m = re.man >>> m.groups( ('username',  groupdict()  A groupdict() expression	tch(r'(\w+)@(\w+)\. ) 'hackerrank', 'com' ssion returns a dictionar	(\w+)','username@h	ackerrank.com')		
A groups() expression  Code  >>> import re >>> m = re.mat >>> m.groups( ('username',  groupdict()  A groupdict() express by the subgroup na	tch(r'(\w+)@(\w+)\.) 'hackerrank', 'com' ssion returns a dictionar	(\w+)','username@h	nackerrank.com') ned subgroups of the match, keye		
A groups() expression  Code  >>> import re >>> m = re.mat >>> m.groups( ('username',  groupdict()  A groupdict() express by the subgroup na  Code  >>> m = re.mat	tch(r'(\w+)@(\w+)\. ) 'hackerrank', 'com' ssion returns a dictionar	(\w+)','username@h ')  y containing all the nam  @(?P <website>\w+)\</website>	nackerrank.com') ned subgroups of the match, keye		
A groups() expression  Code  >>> import re >>> m = re.mat >>> m.groups( ('username',  groupdict()  A groupdict() express by the subgroup na  Code  >>> m = re.mat P <extension>\ &gt;&gt;&gt; m.groupdid</extension>	tch(r'(\w+)@(\w+)\.) 'hackerrank', 'com' ssion returns a dictionar me.  tch(r'(?P <user>\w+) w+)','myname@hacke</user>	(\w+)','username@h ')  y containing all the nam @(?P <website>\w+)\ rrank.com')</website>	nackerrank.com')  ned subgroups of the match, keye		

Author	DOSHI
Difficulty	Easy
Max Score	20
Submitted By	8161
NEED HELP?	
View discussions	
☐ View editorial	
$oldsymbol{\Phi}$ View top submissions	
RATE THIS CHALLENGE	
公 公 公 公 公	
MORE DETAILS	
<b>ل</b> Download sample test cases	
Suggest Edits	
f 💆 in	

## Task You are given a string S. Your task is to find the first occurrence of an alphanumeric character in $m{S}$ (read from left to right) that has consecutive repetitions. Input Format A single line of input containing the string S. Constraints 0 < len(S) < 100**Output Format** Print the first occurrence of the repeating character. If there are no repeating characters, print -1. Sample Input ..12345678910111213141516171820212223 Sample Output 1 **Explanation** .. is the first repeating character, but it is not alphanumeric. 1 is the first (from left to right) alphanumeric repeating character of the string in the substring 111.







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