Group(), Groups() & Groupdict()

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

Try the next challenge | Try a Random Challenge

Problem Submissions Leaderboard Editorial

group()

A group() expression returns one or more subgroups of the match.

Code

```
>>> import re
>>> m = re.match(r'(\w+)@(\w+)\.(\w+)','username@hackerrank.com')
>>> m.group(0)     # The entire match
'username@hackerrank.com'
>>> m.group(1)     # The first parenthesized subgroup.
'username'
>>> m.group(2)     # The second parenthesized subgroup.
'hackerrank'
>>> m.group(3)     # The third parenthesized subgroup.
'com'
>>> m.group(1,2,3)  # Multiple arguments give us a tuple.
('username', 'hackerrank', 'com')
```

groups()

A groups() expression returns a tuple containing all the subgroups of the match.

Code

```
>>> import re
>>> m = re.match(r'(\w+)@(\w+)\.(\w+)','username@hackerrank.com')
>>> m.groups()
('username', 'hackerrank', 'com')
```

groupdict()

A groupdict() expression returns a dictionary containing all the named subgroups of the match, keyed by the subgroup name.

Code

```
>>> m = re.match(r'(?P<user>\w+)@(?P<website>\w+)\.(?P<extension>\w+)','myname@hackerrank.com')
>>> m.groupdict()
{'website': 'hackerrank', 'user': 'myname', 'extension': 'com'}
```

Task

You are given a string S.

Your task is to find the first occurrence of an alphanumeric character in 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`.

Change Theme Language Python 3

```
# Enter your code here. Read input from STDIN. Print output to STDOUT
import re

pattern = r"([a-zA-Z0-9])\1"
s = input()
m = re.search(pattern, s)
if m:
    print(m.group(1))
else:
    print(-1)


```

EMACS Line: 12 Col: 1

Upload Code as File | Test against custom input | Run Code | Submit Code

You have earned 20.00 points!
80/115 challenges solved.

70%

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

- Test case 0
- Test case 1
- Test case 2
- Test case 3
- Test case 4
- Test case 5

Compiler Message

```
Success
```

Input (stdin) Download

```
1 12345678910111213141516171820212223
```

Expected Output Download

```
1 1
```