

# Josh Solution

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

import re

x = input()

m = re.match(r'([a-zA-Z0-9])1\*',x)

y =''

y = m.group(1)

if y == '':

    z = -1

else:

    z = m.group(1)

print(z)

# Other Solution

A pythonic way to do something like this is to use zip() to make consecutive pairs of tuples (like (1, 2), (2, 3) ... and then just return the first one where both are equal and alphanumeric.

next() takes an optional parameter for what to return when there's nothing left, to which here you can pass -1. [isalnum()](https://docs.python.org/3/library/stdtypes.html" \l "str.isalnum) can be used to test is a string is alphanumeric:

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

import re

def firstRepeatedChar(s):

    return next((j for j, i in zip(s, s[1:]) if j == i and j.isalnum()), -1)

print(firstRepeatedChar(input()))

firstRepeatedChar("..12345678910111213141516171820212223")

# 1

firstRepeatedChar("^^cmmit^^")

# 'm'

firstRepeatedChar("^^cmit^^")

# -1