In [1]:
```
#question 1
```

In [27]:
```
import re
s=input("Enter the string: ")
pat=re.compile(r'^a[a-zA-Z]{3}z$')
matched=pat.search(s)
if matched !=None:
    print("serach successful")
else:
    print("search unsucessful")
```

```
Enter the string: hello
search unsucessful
```

In [ ]:

In [3]:
```
#question 2
```

In [26]:
```
import re
def text_match(text):
        patterns = '^a(b*)$'
        if re.search(patterns,  text):
                return ('Match found!')
        else:
                return('Not matched!')
print(text_match("ac"))
print(text_match("a"))
```

```
Not matched!
Match found!
```

In [ ]:

In [ ]:
```
#question 3
```

In [25]:
```
import re
def text_match(text):
        patterns = 'ab+?'
        if re.search(patterns,  text):
                return('Match found!')
        else:
                return('Not matched!')

print(text_match("abc"))
print(text_match("ab"))
```

```
Match found!
Match found!
```

In [ ]:

In [24]:
```
import re
def text_match(text):
    patterns = ' '
    if re.search(patterns, text):
        return ('Match found!')
    else:
        return( 'Not matched!')
print(text_match("aGo"))
print(text_match("Go b"))
```

```
print(text_match("Go c"))
print(text_match("Go ab"))
```

```
Not matched!
Match found!
Match found!
Match found!
```

In [ ]:

In [23]:
```
import re
def text_match(text):
        patterns = 'ab{3}?'
        if re.search(patterns,  text):
                return ('Match found!')
        else:
                return('Not matched!')

print(text_match ("bbba"))
print(text_match("aabbbbbc"))
print(text_match("abbb"))
print(text_match("cbbbba"))
```

```
Not matched!
Match found!
Match found!
Not matched!
```

In [ ]:

In [22]:
```
import re
def text_match(text):
        patterns = 'ab{2,3}'
        if re.search(patterns,  text):
                return ('Match found!')
        else:
                return('Not matched!')
print(text_match("abbbbc"))
print(text_match("abbc"))
print(text_match("abbbbbbbbc"))
print(text_match("abbbc"))
```

```
Match found!
Match found!
Match found!
Match found!
```

In [ ]:

In [21]:
```
import re
def text_match(text):
        patterns = '^[a-z]+_[a-z]+$'
        if re.search(patterns,  text):
                return ('Match found!')
        else:
                return('Not matched!')
print(text_match("Aaabb_abbc"))
print(text_match("ab_cbbc"))
print(text_match("aabb_Abbc"))
```

```
Not matched!
Match found!
Not matched!
```

In [20]:
```
import re
def text_match(text):
        patterns = 'ab{2,3}'
        if re.search(patterns,  text):
                return ('Match found!')
        else:
                return('Not matched!')
print(text_match("acdeb"))
print(text_match("aefdb"))
print(text_match("acb"))
print(text_match("axefcb"))
```

```
Not matched!
Not matched!
Not matched!
Not matched!
```

In [ ]:
```
```

In [19]:
```
import re
def text_match(text):
        patterns = '^\w+'
        if re.search(patterns,  text):
                return ('Match found!')
        else:
                return('Not matched!')

print(text_match(" The science is best part of the nature."))
print(text_match("The science is best part of the nature."))
```

```
Not matched!
Match found!
```

In [30]:
```
import re
def text_match(text):
        patterns = '^[a-zA-Z0-9_]*$'
        if re.search(patterns,  text):
                return ('Match found!')
        else:
                return('Not matched!')

print(text_match("The science is best part of the nature."))
print(text_match("Fusion_Series"))
```

```
Not matched!
Match found!
```

In [31]:
```
import re
def match_num(string):
    text = re.compile(r"^5")
    if text.match(string):
        return False
    else:
        return True
print(match_num('9-123456'))
print(match_num('7-654321'))
```

```
True
True
```

In [33]:
```
import re
ip = "45.112.30.154"
string = re.sub('\.[0]*', '.', ip)
print(string)
```

```
45.112.30.154
```

In [34]:
```
import re
patterns = [ 'fox', 'dog', 'horse' ]
text =('The quick brown fox jumps over the lazy dog.')
for pattern in patterns:
    print('Searching for "%s" in "%s" ->' % (pattern, text),)
    if re.search(pattern,  text):
        print('Match!')
    else:
        print('Not Matched!')
```

```
Searching for "fox" in "The quick brown fox jumps over the lazy dog." ->
Match!
Searching for "dog" in "The quick brown fox jumps over the lazy dog." ->
Match!
Searching for "horse" in "The quick brown fox jumps over the lazy dog." ->
Not Matched!
```

In [35]:
```
import re
pattern = 'fox'
text = ('The quick brown fox jumps over the lazy dog.')
match = re.search(pattern, text)
s = match.start()
e = match.end()
print('Found "%s" in "%s" from %d to %d ' % \
    (match.re.pattern, match.string, s, e))
```

```
Found "fox" in "The quick brown fox jumps over the lazy dog." from 16 to 19
```

In [39]:
```
import re

pattern = 'exercises'
text = ('Python exercises, PHP exercises, C# exercises')
for match in re.findall(pattern, text):
    print('Found "%s"' % match)
```

```
Found "exercises"
Found "exercises"
Found "exercises"
```

In [ ]:
```
```

In [40]:
```
import re

pattern = 'exercises'
text = ('Python exercises, PHP exercises, C# exercises')
for match in re.finditer(pattern, text):
    s = match.start()
    e = match.end()
    print('Found "%s" at %d:%d' % (text[s:e], s, e))
```

```
Found "exercises" at 7:16
Found "exercises" at 22:31
Found "exercises" at 36:45
```

In [43]:
```
import re
def change_date_format(dt):
        return re.sub(r'(\d{4})-(\d{1,2})-(\d{1,2})', '\\3-\\2-\\1', dt)
dt1 = "2025-10-03"
print("Original date in YYY-MM-DD Format: ",dt1)
print("New date in DD-MM-YYYY Format: ",change_date_format(dt1))
```

```
Original date in YYY-MM-DD Format:  2025-10-03
New date in DD-MM-YYYY Format:  03-10-2025
```

In [45]:
```
import re
text =
list = re.findall("[ae]\w+", text)
print(list)
```

```
['engineers', 'accountants', 'aeronautical', 'enigneers']
```

In [50]:
```
import re
num_list = map(int, occ)
occ = re.findall("\d+", "They are 143780 people in my city out of which 3567 are i
print(max(num_list))
```

```
143780
```

In [53]:
```
import re
def capital_words_spaces(str1):
    return re.sub(r"(\w)([A-Z])", r"\1 \2", str1)

print(capital_words_spaces("India"))
print(capital_words_spaces("India Country"))
print(capital_words_spaces("India Country Is Great"))
```

```
India
India Country
India Country Is Great
```

In [54]:
```
import re
def text_match(text):
        patterns = '[A-Z]+[a-z]+$'
        if re.search(patterns, text):
                return ('Matched!')
        else:
                return('Not matched!')

print(text_match("Vehana"))
print(text_match("JoShInI"))
```

```
Matched!
Not matched!
```

In [58]:
```
import re
print(re.findall("#(\w+)", input("enter your name")))
```

```
enter your namesirisha
[]
```

In [61]:
```
import re
text = "@Jags123456 Bharat band on 28??<ed><U+00A0><U+00BD><ed><U+00B8><U+0082>Thos
newtext = re.sub(r"^[^\w$€]+|[^\w$€]+$", "", text)
print(newtext)
```

```
Jags123456 Bharat band on 28??<ed><U+00A0><U+00BD><ed><U+00B8><U+0082>Those who  a
re protesting #demonetization  are all different party leaders
```

In [62]:
```
import re
text = ('Python Exercises, PHP exercises.')
print(re.sub("[ ,.]", ":", text))
```

```
Python:Exercises::PHP:exercises:
```

In [ ]:
```

```

In [ ]:
```

```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: