2) Write a Python program to check that a string contains only a certain set of characters (in this case a-z, A-Z and 0-9).

```
import re


str1=input()
str2=input()

charRe = re.compile(r'[^a-zA-Z0-9]')

string = charRe.search(str1)
print(not bool(string))

string2 = charRe.search(str2)
print(not bool(string2))
```

```
Abh53f
@ghj
True
False
```

3) Write a Python program that matches a string that has an a followed by zero or more b's

```
import re

ip=input()
pattern= re.compile(r'ab*')

if re.fullmatch(pattern, ip):
    print("True")
    # occurrences=[]
    # for match in pattern.finditer(ip):
    #     start = match.start()
    #     end = match.end()
    #     occurrences.append((start, end))

    # print(occurrences)

else:
   print("False")
```

```
a
True
```

4) Write a Python program that matches a string that has an a followed by one or more b's

```
import re

ip=input()
pattern=r'ab+'

if re.fullmatch(pattern, ip):
    print("True")
else:
   print("False")
```

```
a
False
```

5) Write a Python program that matches a string that has an a followed by two to three 'b'.

```
import re

ip=input()

pattern=r'ab{2,3}'

if re.fullmatch(pattern, ip):
    print("True")
else:
   print("False")
```

```
abbb
True
```

6)Write a Python program that matches a string that has an 'a' followed by anything, ending in 'b'.

```
import re

ip=input()
pattern=r'a.*b$'

if re.fullmatch(pattern, ip):
    print("True")

else:
   print("False")
```

```
ajkfjvkjkb
True
```

7)Write a Python program to match a string that contains only upper and lowercase letters, numbers, and underscores.

```
import re

ip=input()

pattern=r'^[a-zA-Z0-9_]*$'

if re.fullmatch(pattern, ip):
    print("True")

else:
   print("False")
```

```
Ab1_
True
```

8) Write a Python program to check for a number at the end of a string.

```
import re

ip=input()
pattern=r'.*[0-9]$'



if re.fullmatch(pattern, ip):
    print("True")
else:
   print("False")
```

```
hghygdu12
True
```

9) Write a Python program to find the occurrence and position of the substrings within a string.

```
import re

def find_occurrences_positions(main_string, substring):
    occurrences = []
    cnt=0
    pattern = re.compile(re.escape(substring))

    for match in pattern.finditer(main_string):
        start = match.start()
        end = match.end()
        occurrences.append((start, end))
        cnt+=1
        #ls.append(str(match))

    return [occurrences,"Total Occurrences : {}".format(cnt)]

print(find_occurrences_positions("abbbabbba","bbb"))
```

```
[[(1, 4), (5, 8)], 'Total Occurrences : 2']
```

Try the following scripts:

```
import re
pattern = "Py.*n"
count = 0
text = ["Python coding", "Pyt3on","Java", "Py45n", "Py@#n","Pyn"]
for i in text:
    if(re.findall(pattern, i)):
           count+=1
print("Matches found:", count)
```

```
Matches found: 5
```

```
import re
pattern = "Py.+n"
count = 0
text = ["Python coding", "Pyt3on","Java", "Py45n", "Py@#n","Pyn"]
for i in text:
    if(re.findall(pattern, i)):
        count+=1

print("Matches found:", count)
```

```
Matches found: 4
```

```
import re
pattern = "Python"
replacement = "Java"
text = "Python is fun"
substituted_text = re.sub(pattern, replacement, text)
print("Substituted text:", substituted_text)
```

```
Substituted text: Java is fun
```

Generators:

```
#Generator
def is_even():
  num=0
  while True:
    if num%2==0:
      yield num
    num+=1

even=is_even()
for _ in range(10):
    print(next(even))
```

```
0
2
4
6
8
10
12
14
16
18
```

Iterator:

```
class Fibbo:
    def __init__(self, limit):
        self.limit = limit
        self.a, self.b = 0, 1

    def   iter  (self):
        return self

    def __next__(self):
        if self.a > self.limit:
            raise StopIteration
        result = self.a
        self.a, self.b = self.b, self.a + self.b
        return result


fibbo = Fibbo(100)

for num in fibbo:
    print(num)
```

```
0
1
1
2
3
5
8
13
21
34
55
89
```

✓ 0s completed at 10:22 AM ● ×