1. Python – Least Frequent Character in String

In [2]:
```
from collections import Counter

test_str = "GoodsforGood"

print ("The original string is : " + test_str)


res = Counter(test_str)
res = min(res, key = res.get)

print ("The minimum of all characters in Goodsforgood is : " + str(res))
```

```
The original string is : GoodsforGood
The minimum of all characters in Goodsforgood is : s
```

2. Python | Maximum frequency character in String

In [3]:
```
from collections import Counter

test_str = "GoodsforGood"

print ("The original string is : " + test_str)


res = Counter(test_str)
res = max(res, key = res.get)

print ("The maximum of all characters in GoodsforGood is : " + str(res))
```

```
The original string is : GoodsforGood
The maximum of all characters in GoodsforGood is : o
```

3. Python | Program to check if a string contains any special character

In [4]:
```
n="Goods$For$Good"
n.split()
c=0
s='[@_!#$%^&*()<>?/\|}{~:]'
for i in range(len(n)):

    if n[i] in s:
      c+=1


if c:
    print("string is not accepted")
else:
    print("string accepted")
```

```
string is not accepted
```

4. Generating random strings until a given string is generated

In [5]:
```
import string
import random
import time

possibleChar = string.ascii_lowercase + string.digits + string.ascii_uppercase + ' .,!?;:'

t = "test"

attemptThis = ''.join(random.choice(possibleChar) for i in range(len(t)))
attemptNext = ''

done= False
iteration = 0

while done == False:
    print(attemptThis)

    attemptNext = ''
    done = True

    for i in range(len(t)):
        if attemptThis[i] != t[i]:
            done = False
            attemptNext += random.choice(possibleChar)
        else:
            attemptNext += t[i]

    iteration = iteration + 1
    attemptThis = attemptNext
    time.sleep(0.1)

print("Target matched after ",iteration," iterations")
```

```
WrI6
pWc7
o7Kh
ywGS
ol18
qdRa
;r2U
.Kz
;1hr
!Vpg
vAye
S.f,
vAye
J!sS
G1su
INsu
zRsG
Xcs;
gDs;
dosy
?2s7
sksF
0xsj
Bjs2
KXsG
7es;
5es1
uesN
1es.
eest
Yest
7est
Mest
vest
Cest
Hest
!est
Eest
Aest
:est
aest
Best
3est
Dest
Eest
fest
Mest
cest
Lest
aest
qest
test
Target matched after  52  iterations
```

5. Find words which are greater than given length k

In [6]:
```
def string_k(k, str):

    string = []

    text = str.split(" ")

    for x in text:

        if len(x) > k:

            string.append(x)

    return string


k = 3
str ="goods for good"
print(string_k(k, str))
```

```
['goods', 'good']
```

6. Python program for removing i-th character from a string

In [7]:
```
def remove(string, i):

    a = string[ : i]

    b = string[i + 1: ]

    return a + b

if __name__ == '__main__':

    string = "goodsFORgood"

    i = 5

    print(remove(string, i))
```

```
goodsORgood
```

7. Python program to split and join a string

In [8]:
```
def split_string(string):

    list_string = string.split(' ')

    return list_string

def join_string(list_string):

    string = '-'.join(list_string)

    return string

if __name__ == '__main__':
    string = 'Goods for Good'

    list_string = split_string(string)
    print(list_string)


    new_string = join_string(list_string)
    print(new_string)
```

```
['Goods', 'for', 'Good']
Goods-for-Good
```

8. Python | Check if a given string is binary string or not

In [9]:
```
def check(string) :

    p = set(string)

    s = {'0', '1'}

    if s == p or p == {'0'} or p == {'1'}:
        print("Yes")
    else :
        print("No")

if __name__ == "__main__" :

    string = "101010000111"

    check(string)
```

```
Yes
```

9. Python program to find uncommon words from two Strings

In [10]:
```
def UncommonWords(A, B):

    count = {}

    for word in A.split():
        count[word] = count.get(word, 0) + 1

    for word in B.split():
        count[word] = count.get(word, 0) + 1

    return [word for word in count if count[word] == 1]

A = "Goods for Good"
B = "Learning from program Good teacher"

print(UncommonWords(A, B))
```

```
['Goods', 'for', 'Learning', 'from', 'program', 'teacher']
```

10. Python – Replace duplicate Occurrence in String.

In [17]:
```
my_str = 'Sam is the best . Sam loves to cook. Sam and Will cook together'
print("The string is : ")
print(my_str)
replace_dict = {'Sam' : 'She' }
my_list = my_str.split(' ')
my_result = ' '.join([replace_dict.get(val) if val in replace_dict.keys() and my_list.index(val) != idx else val fo
print("The string after replacing with values is : ")
print(my_result)
```

```
The string is :
Sam is the best . Sam loves to cook. Sam and Will cook together
The string after replacing with values is :
Sam is the best . She loves to cook. She and Will cook together
```

In [ ]: