1. Python – Least Frequent Character in String
2. Python | Maximum frequency character in String
3. Python | Program to check if a string contains any special character
4. Generating random strings until a given string is generated
5. Find words which are greater than given length k
6. Python program for removing i-th character from a string
7. Python program to split and join a string
8. Python | Check if a given string is binary string or not
9. Python program to find uncommon words from two Strings
10. Python – Replace duplicate Occurrence in String

Solution

Here are the solutions to the listed Python problems:

### 1. Python – Least Frequent Character in String

```python

from collections import Counter

def least\_frequent\_char(s):

freq = Counter(s)

min\_freq = min(freq.values())

least\_freq\_chars = [char for char in freq if freq[char] == min\_freq]

return least\_freq\_chars

# Example usage

s = "aabbccddeeffg"

print(least\_frequent\_char(s))

```

### 2. Python | Maximum Frequency Character in String

```python

from collections import Counter

def max\_frequency\_char(s):

freq = Counter(s)

max\_freq = max(freq.values())

max\_freq\_chars = [char for char in freq if freq[char] == max\_freq]

return max\_freq\_chars

# Example usage

s = "aabbccddeeffg"

print(max\_frequency\_char(s))

```

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

```python

import re

def contains\_special\_characters(s):

return bool(re.search(r'[^a-zA-Z0-9]', s))

# Example usage

s = "Hello@World!"

print(contains\_special\_characters(s)) # True

```

### 4. Generating random strings until a given string is generated

```python

import random

import string

def generate\_random\_string(length):

return ''.join(random.choices(string.ascii\_letters + string.digits, k=length))

def generate\_until\_target(target):

attempts = 0

while True:

random\_string = generate\_random\_string(len(target))

attempts += 1

if random\_string == target:

return attempts

# Example usage

target = "abc123"

print(generate\_until\_target(target))

```

### 5. Find words which are greater than a given length k

```python

def find\_long\_words(s, k):

words = s.split()

long\_words = [word for word in words if len(word) > k]

return long\_words

# Example usage

s = "Find all words which are longer than a certain length"

k = 4

print(find\_long\_words(s, k))

```

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

```python

def remove\_ith\_char(s, i):

return s[:i] + s[i+1:]

# Example usage

s = "Hello World"

i = 4

print(remove\_ith\_char(s, i)) # "Hell World"

```

### 7. Python program to split and join a string

```python

def split\_and\_join(s, delimiter):

words = s.split()

return delimiter.join(words)

# Example usage

s = "Python is awesome"

delimiter = "-"

print(split\_and\_join(s, delimiter)) # "Python-is-awesome"

```

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

```python

def is\_binary\_string(s):

return set(s).issubset({'0', '1'})

# Example usage

s = "101010"

print(is\_binary\_string(s)) # True

```

### 9. Python program to find uncommon words from two Strings

```python

def find\_uncommon\_words(s1, s2):

set1 = set(s1.split())

set2 = set(s2.split())

uncommon\_words = list(set1.symmetric\_difference(set2))

return uncommon\_words

# Example usage

s1 = "apple banana mango"

s2 = "banana orange apple"

print(find\_uncommon\_words(s1, s2))

```

### 10. Python – Replace duplicate Occurrence in String

```python

def replace\_duplicate\_occurrence(s, replace\_char='\*'):

words = s.split()

seen = set()

result = []

for word in words:

if word in seen:

result.append(replace\_char)

else:

result.append(word)

seen.add(word)

return ' '.join(result)

# Example usage

s = "this is a test this is only a test"

print(replace\_duplicate\_occurrence(s))

```

These code snippets solve the given problems in Python. Adjust the examples according to your needs.