

1. Dice probability distribution simulation (20 points) You roll the dice 10,000 times. Each time you get one of six sides with a probability of 1/6. You need to generate the total number of occurrences for each side, store them in a dictionary. For example: If you roll the dice 10 times. Assume the number of occurrences for each side is this: Sides 1 2 3 4 5 6
occurrences 3 0 2 1 1 3 Then the output is: {1:3, 2:0, 3:2, 4:1, 5:1, 6:3} Hint: This rolling is equal to choosing a random number in [1,2,3,4,5,6]. You need to use random.randint(start, stop) to generate an integer between 1 and 6.

In [5]:

```
import random

occurrences = {1: 0, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0}

for i in range(10000):
    roll = random.randint(1, 6)
    occurrences[roll] += 1

print(occurrences)
```

{1: 1696, 2: 1641, 3: 1666, 4: 1694, 5: 1626, 6: 1677}

1. Sentence comparison (20 points) You receive two sentences as inputs from the keyboard and compare them. The output is the number of common unique characters in two sentences. It should be case insensitive. For example: sentence 1: "I like Python programming" sentence 2: "I like java" Then the output is: 6. (they are: i, l, k, e, a, and space). Hint: To convert a string to a lowercase string, use s.lower()

```
x = 'HELLO'

print(x.lower())
```

Hello

In [4]:

```
sentence1 = input("Enter the first sentence: ")
sentence2 = input("Enter the second sentence: ")

sentence1 = sentence1.lower()
sentence2 = sentence2.lower()

unique_chars1 = set(sentence1)
unique_chars2 = set(sentence2)

common_chars = unique_chars1.intersection(unique_chars2)
num_common_chars = len(common_chars)
print("The number of common unique characters is:", num_common_chars)
```

```
Enter the first sentence: i love dogs
Enter the second sentence: dogs are cute
The number of common unique characters is: 6
```

1. Sentence split (30 points) You receive one sentence and two delimiters from the keyboard. Then you separate all strings in that sentence. For example: sentence: "I like computer

programming, including Python, Java, and C/C++" delimiters: , / Then the output list will be: ['I like computer programming', 'including Python', 'Java', 'and C', 'C++'] Hint: One suggestion is to use variable length arguments. You can also consider split() function for strings.

```
In [3]: sentence = input("Enter the sentence: ")
delimiter1, delimiter2 = input("Enter the two delimiters (separated by a space):

sentence = sentence.replace(delimiter1, ',')
sentence = sentence.replace(delimiter2, ',')

split_list = [s.strip() for s in sentence.split(',')])

print(split_list)
```

```
Enter the sentence: i love corgis
Enter the two delimiters (separated by a space): . ,
['i love corgis']
```

1. Sort dictionary (30 points) Write a program that can sort a given dictionary either by key or value in ascending order. The program allows users to choose operations (1: sort by key, 2: sort by value). If the given dictionary is: d = {'x': 7, 'y': 2, 'a': 3, 'm': 2}, the running output is following. please select operation: (1: sort by key, 2: sort by value) 1 a , 3 m , 2 x , 7 y , 2 please select operation: (1: sort by key, 2: sort by value) 2 y , 2 m , 2 a , 3 x , 7

```
In [2]: d = {'x': 7, 'y': 2, 'a': 3, 'm': 2}

while True:
    operation = input("Please select operation: (1: sort by key, 2: sort by value) ")

    if operation == '1':
        sorted_dict = {k: d[k] for k in sorted(d)}
    elif operation == '2':
        sorted_dict = {k: v for k, v in sorted(d.items(), key=lambda item: item[1])}
    else:
        print("Invalid operation. Please try again.")
        continue

    for k, v in sorted_dict.items():
        print(k, ", ", v)

    choice = input("Do you want to continue? (y/n) ")
    if choice.lower() != 'y':
        break
```

```
Please select operation: (1: sort by key, 2: sort by value) 1
a , 3
m , 2
x , 7
y , 2
Do you want to continue? (y/n) n
```

In []: