Task Level (Beginner):

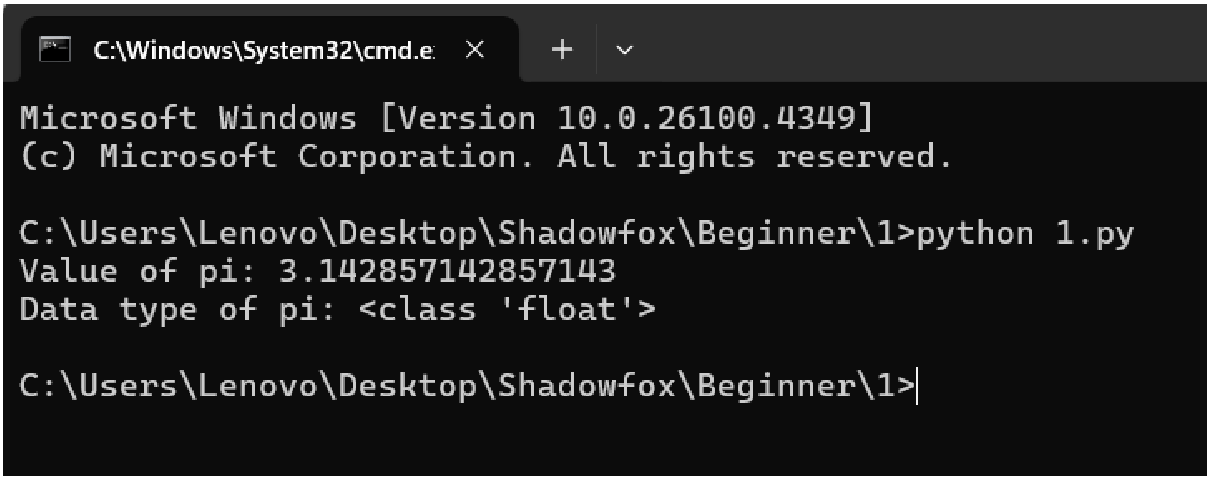
1. Variables

1. Create a variable named pi and store the value 22/7 in it. Now check the data type of this variable.

pi = 22 / 7

print("Value of pi:", pi)

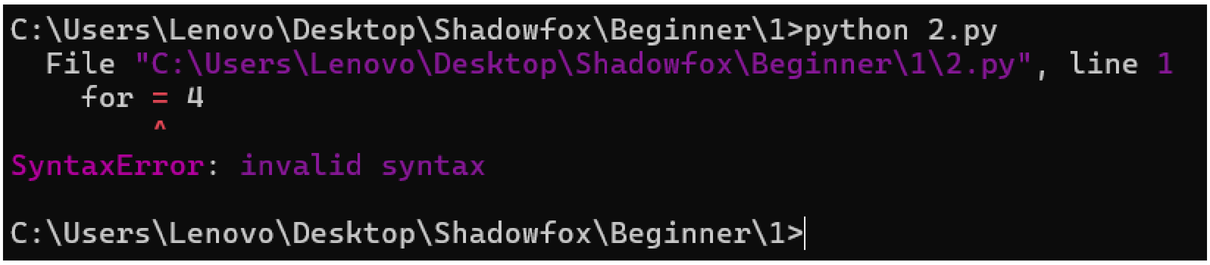
print("Data type of pi:", type(pi))



```
Microsoft Windows [Version 10.0.26100.4349]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Lenovo\Desktop\Shadowfox\Beginner\1>python 1.py
Value of pi: 3.142857142857143
Data type of pi: <class 'float'>

C:\Users\Lenovo\Desktop\Shadowfox\Beginner\1>
```

2. Create a variable called for and assign it a value 4. See what happens and find out the reason behind the behavior that you see.

for = 4



```
C:\Users\Lenovo\Desktop\Shadowfox\Beginner\1>python 2.py
  File "C:\Users\Lenovo\Desktop\Shadowfox\Beginner\1\2.py", line 1
    for = 4
        ^
SyntaxError: invalid syntax

C:\Users\Lenovo\Desktop\Shadowfox\Beginner\1>
```

This throws a SyntaxError because for is a reserved keyword in Python. It is used in for loops, so you can't use it as a variable name.

To fix this, use a different name like:

for_value = 4

3. Store the principal amount, rate of interest, and time in different variables and then calculate the Simple Interest for 3 years. Formula: Simple Interest = P x R x T / 100

Assign values

principal = 10000 # Example value

rate = 5   # Interest rate in percent

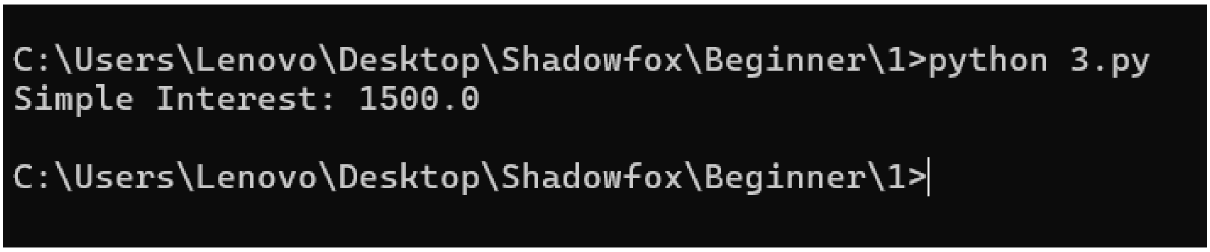
time = 3   # Time in years

# Calculate simple interest

simple_interest = (principal * rate * time) / 100

# Output the result

print("Simple Interest:", simple_interest)



```
C:\Users\Lenovo\Desktop\Shadowfox\Beginner\1>python 3.py
Simple Interest: 1500.0

C:\Users\Lenovo\Desktop\Shadowfox\Beginner\1>
```

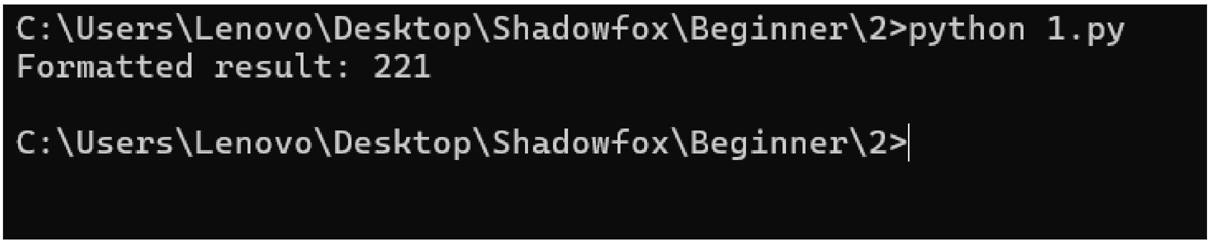
2. Numbers

1. Write a function that takes two arguments, 145 and 'o', and uses the `format` function to return a formatted string. Print the result. Try to identify the representation used.

number = 145

formatted = format(number, 'o')

print("Formatted result:", formatted)

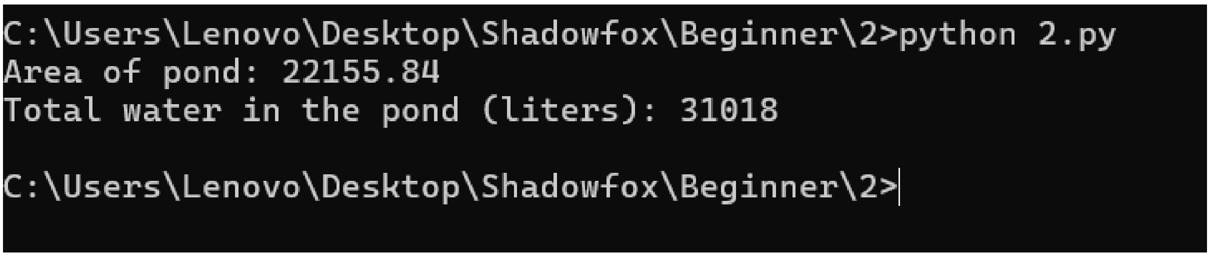


```
C:\Users\Lenovo\Desktop\Shadowfox\Beginner\2>python 1.py
Formatted result: 221

C:\Users\Lenovo\Desktop\Shadowfox\Beginner\2>
```

The 'o' format code converts a number to octal (base 8).
So, 145 in decimal becomes 221 in octal.

2. In a village, there is a circular pond with a radius of 84 meters. Calculate the area of the pond using the formula: Circle Area = π r^2. (Use the value 3.14 for π) Bonus Question: If there is exactly 1.4 liters of water in a square meter, what is the total amount of water in the pond? Print the answer without any decimal point in it. Hint: Circle Area = π r^2 Water in the pond = Pond Area Water per Square Meter

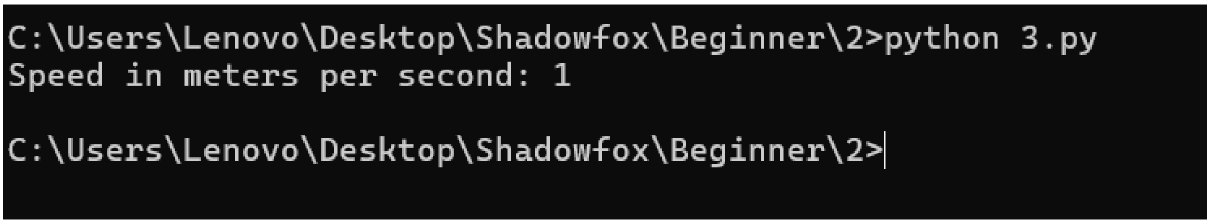
```
pi = 3.14

radius = 84

area = pi * radius ** 2

water_per_sq_meter = 1.4

total_water = area * water_per_sq_meter

print("Area of pond:", area)

print("Total water in the pond (liters):", int(total_water))
```



```
C:\Users\Lenovo\Desktop\Shadowfox\Beginner\2>python 2.py
Area of pond: 22155.84
Total water in the pond (liters): 31018

C:\Users\Lenovo\Desktop\Shadowfox\Beginner\2>
```

3. If you cross a 490meterlong street in 7 minutes, calculate your speed in meters per second. Print the answer without any decimal point in it. Hint: Speed = Distance / Time

```
distance = 490

time_seconds = 7 * 60

speed = distance / time_seconds

print("Speed in meters per second:", int(speed))
```



```
C:\Users\Lenovo\Desktop\Shadowfox\Beginner\2>python 3.py
Speed in meters per second: 1

C:\Users\Lenovo\Desktop\Shadowfox\Beginner\2>
```

3. List

1. You have a list of superheroes representing the Justice League. justice_league = ["Superman", "Batman", "Wonder Woman", "Flash", "Aquaman", "Green Lantern"] Perform the following tasks: 1. Calculate the number of members in the Justice League.

2. Batman recruited Batgirl and Nightwing as new members. Add them to your list.

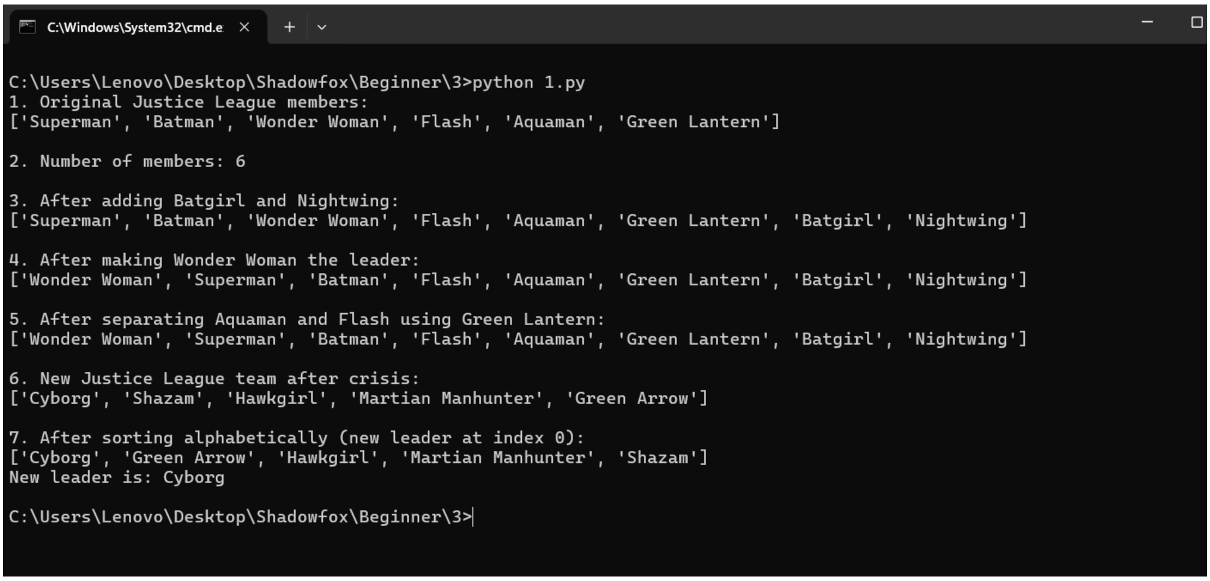
3. Wonder Woman is now the leader of the Justice League. Move her to the beginning of the list.
4. Aquaman and Flash are having conflicts, and you need to separate them. Choose either "Green Lantern" or "Superman" and move them in between Aquaman and Flash.

5. The Justice League faced a crisis, and Superman decided to assemble a new team. Replace the existing list with the following new members: "Cyborg", "Shazam", "Hawkgirl", "Martian Manhunter", "Green Arrow".

6. Sort the Justice League alphabetically. The hero at the 0th index will become the new leader. (BONUS: Can you predict who the new leader will be?) Your task is to write Python code to perform these operations on the "justice_league" list. Display the list at each step to observe the changes.

```
# Step 1: Initial Justice League list
justice_league = ["Superman", "Batman", "Wonder Woman", "Flash", "Aquaman", "Green Lantern"]
print("1. Original Justice League members:")
print(justice_league)

# Step 2: Calculate number of members
num_members = len(justice_league)
print("\n2. Number of members:", num_members)

# Step 3: Add Batgirl and Nightwing
justice_league.append("Batgirl")
justice_league.append("Nightwing")
print("\n3. After adding Batgirl and Nightwing:")
print(justice_league)

# Step 4: Move Wonder Woman to the beginning (as leader)
justice_league.remove("Wonder Woman")
justice_league.insert(0, "Wonder Woman")
print("\n4. After making Wonder Woman the leader:")
print(justice_league)

# Step 5: Separate Aquaman and Flash by inserting Green Lantern in between
# First, remove Green Lantern so we can reposition it
justice_league.remove("Green Lantern")
```

```
# Get the index of Aquaman and insert Green Lantern after Aquaman

index_of_aquaman = justice_league.index("Aquaman")

justice_league.insert(index_of_aquaman + 1, "Green Lantern")

print("\n5. After separating Aquaman and Flash using Green Lantern:")

print(justice_league)


# Step 6: Crisis - Replace list with a new team

justice_league = ["Cyborg", "Shazam", "Hawkgirl", "Martian Manhunter", "Green Arrow"]

print("\n6. New Justice League team after crisis:")

print(justice_league)


# Step 7: Sort alphabetically and assign leader at index 0

justice_league.sort()

print("\n7. After sorting alphabetically (new leader at index 0):")

print(justice_league)

print("New leader is:", justice_league[0])
```



```
C:\Users\Lenovo\Desktop\Shadowfox\Beginner\3>python 1.py
1. Original Justice League members:
['Superman', 'Batman', 'Wonder Woman', 'Flash', 'Aquaman', 'Green Lantern']

2. Number of members: 6

3. After adding Batgirl and Nightwing:
['Superman', 'Batman', 'Wonder Woman', 'Flash', 'Aquaman', 'Green Lantern', 'Batgirl', 'Nightwing']

4. After making Wonder Woman the leader:
['Wonder Woman', 'Superman', 'Batman', 'Flash', 'Aquaman', 'Green Lantern', 'Batgirl', 'Nightwing']

5. After separating Aquaman and Flash using Green Lantern:
['Wonder Woman', 'Superman', 'Batman', 'Flash', 'Aquaman', 'Green Lantern', 'Batgirl', 'Nightwing']

6. New Justice League team after crisis:
['Cyborg', 'Shazam', 'Hawkgirl', 'Martian Manhunter', 'Green Arrow']

7. After sorting alphabetically (new leader at index 0):
['Cyborg', 'Green Arrow', 'Hawkgirl', 'Martian Manhunter', 'Shazam']
New leader is: Cyborg

C:\Users\Lenovo\Desktop\Shadowfox\Beginner\3>
```

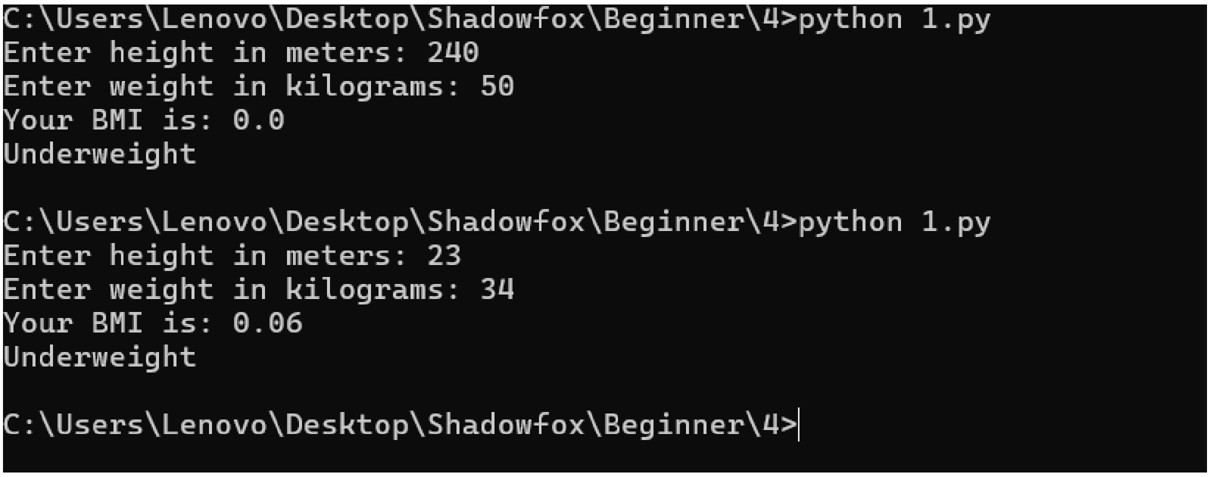
BONUS Answer:

New Leader after sorting alphabetically: Cyborg

4. If Condition

1. Write a program to determine the BMI Category based on user input. Ask the user to: Enter height in meters Enter weight in kilograms Calculate BMI using the formula: BMI = weight / (height)2 Use the following categories: If BMI is 30 or greater, print "Obesity" If BMI is between 25 and 29, print "Overweight" If BMI is between 18.5 and 25, print "Normal" If BMI is less than 18.5, print "Underweight" Example: Enter height in meters: 1.75 Enter weight in kilograms: 70 Output: "Normal"

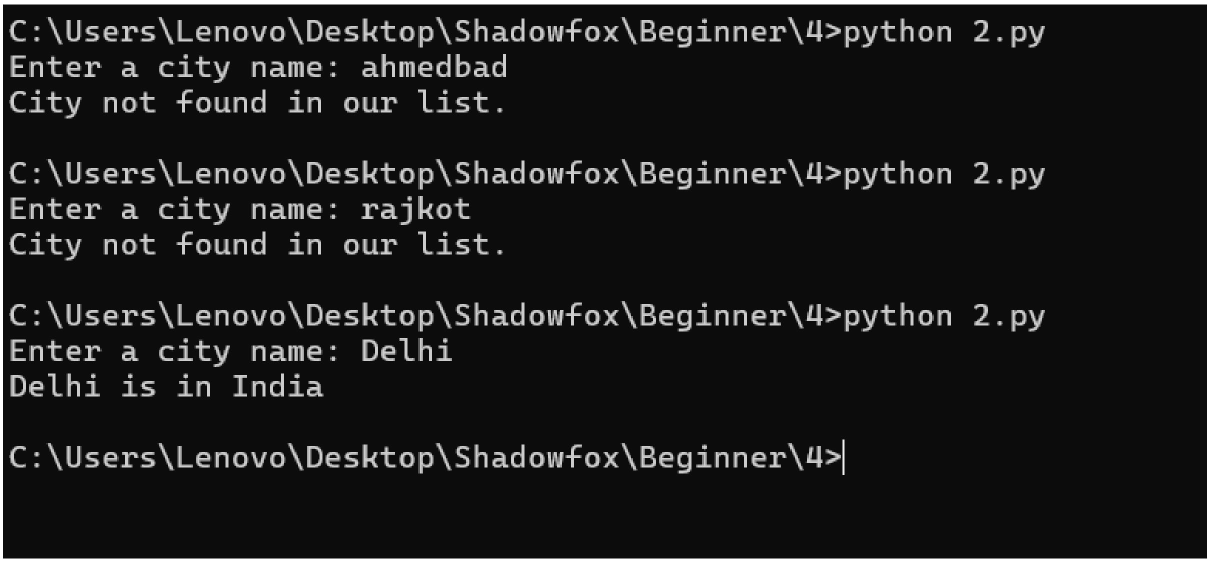
```
# BMI Calculation Program

height = float(input("Enter height in meters: "))

weight = float(input("Enter weight in kilograms: "))



bmi = weight / (height ** 2)



print("Your BMI is:", round(bmi, 2))



if bmi >= 30:

    print("Obesity")

elif bmi >= 25:

    print("Overweight")

elif bmi >= 18.5:

    print("Normal")

else:

    print("Underweight")
```



```
C:\Users\Lenovo\Desktop\Shadowfox\Beginner\4>python 1.py
Enter height in meters: 240
Enter weight in kilograms: 50
Your BMI is: 0.0
Underweight

C:\Users\Lenovo\Desktop\Shadowfox\Beginner\4>python 1.py
Enter height in meters: 23
Enter weight in kilograms: 34
Your BMI is: 0.06
Underweight

C:\Users\Lenovo\Desktop\Shadowfox\Beginner\4>
```

2. Write a program to determine which country a city belongs to. Given list of cities per country: Australia = ["Sydney", "Melbourne", "Brisbane", "Perth"] UAE = ["Dubai", "Abu Dhabi", "Sharjah", "Ajman"] India = ["Mumbai", "Bangalore", "Chennai", "Delhi"] Ask the user to enter a city name and print the corresponding country. Example: Enter a city name: "Abu Dhabi" Output: "Abu Dhabi is in UAE"

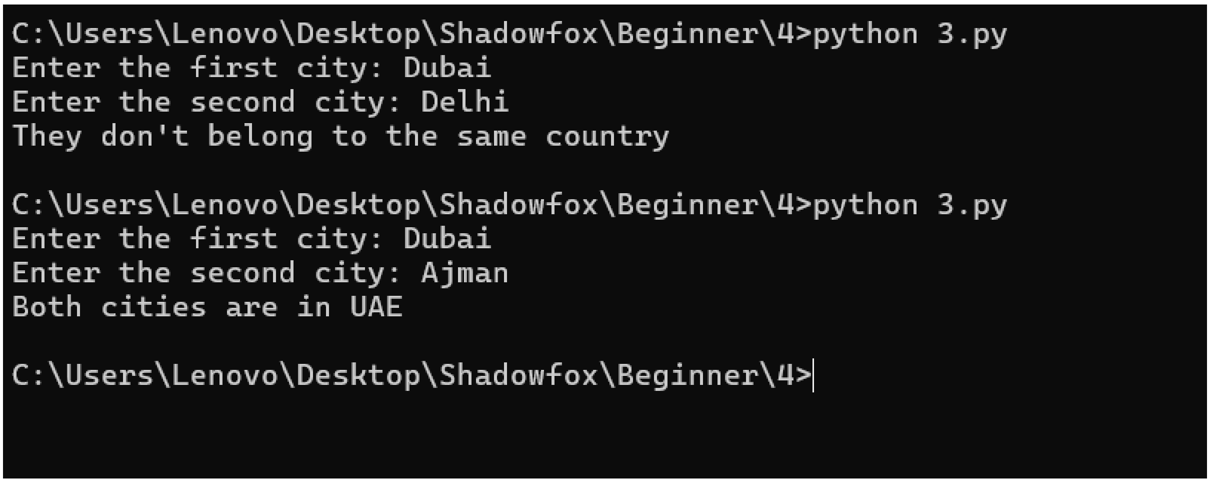
```
# City-Country Finder

city = input("Enter a city name: ")



australia = ["Sydney", "Melbourne", "Brisbane", "Perth"]

uae = ["Dubai", "Abu Dhabi", "Sharjah", "Ajman"]

india = ["Mumbai", "Bangalore", "Chennai", "Delhi"]



if city in australia:

   print(f"{city} is in Australia")

elif city in uae:

   print(f"{city} is in UAE")

elif city in india:

   print(f"{city} is in India")

else:

   print("City not found in our list.")
```



```
C:\Users\Lenovo\Desktop\Shadowfox\Beginner\4>python 2.py
Enter a city name: ahmedbad
City not found in our list.

C:\Users\Lenovo\Desktop\Shadowfox\Beginner\4>python 2.py
Enter a city name: rajkot
City not found in our list.

C:\Users\Lenovo\Desktop\Shadowfox\Beginner\4>python 2.py
Enter a city name: Delhi
Delhi is in India

C:\Users\Lenovo\Desktop\Shadowfox\Beginner\4>
```

3. Write a program to check if two cities belong to the same country. Ask the user to enter two cities and print whether they belong to the same country or not. Example: Enter the first city: "Mumbai" Enter the second city: "Chennai" Output: "Both cities are in India" Example: Enter the first city: "Sydney" Enter the second city: "Dubai" Output: "They don't belong to the same country"

```
# Same Country City Checker

city1 = input("Enter the first city: ")

city2 = input("Enter the second city: ")


australia = ["Sydney", "Melbourne", "Brisbane", "Perth"]

uae = ["Dubai", "Abu Dhabi", "Sharjah", "Ajman"]

india = ["Mumbai", "Bangalore", "Chennai", "Delhi"]


def get_country(city):

    if city in australia:

        return "Australia"

    elif city in uae:

        return "UAE"

    elif city in india:

        return "India"

    else:
        return None


country1 = get_country(city1)
country2 = get_country(city2)


if country1 and country2:
    if country1 == country2:
        print(f"Both cities are in {country1}")
    else:
        print("They don't belong to the same country")
else:
    print("One or both cities not found in our list.")
```



```
C:\Users\Lenovo\Desktop\Shadowfox\Beginner\4>python 3.py
Enter the first city: Dubai
Enter the second city: Delhi
They don't belong to the same country

C:\Users\Lenovo\Desktop\Shadowfox\Beginner\4>python 3.py
Enter the first city: Dubai
Enter the second city: Ajman
Both cities are in UAE

C:\Users\Lenovo\Desktop\Shadowfox\Beginner\4>
```

5. For Loop:

1. Using a for loop, simulate rolling a sixsided die multiple times (at least 20 times).

Count and print the following statistics:

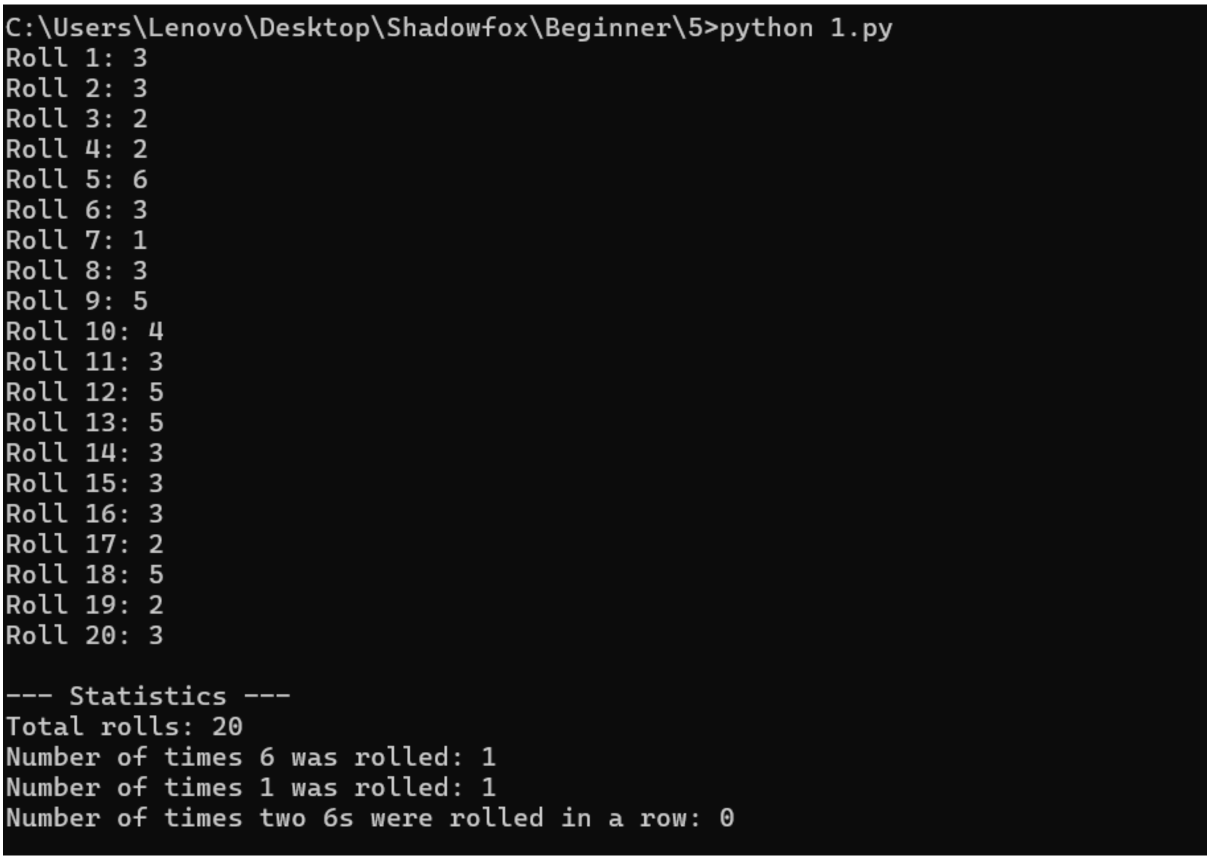
How many times you rolled a 6

How many times you rolled a 1

How many times you rolled two 6s in a row

```
import random


rolls = 20
count_6 = 0
```

```
count_1 = 0
consecutive_6s = 0
previous_roll = 0

for i in range(rolls):
    roll = random.randint(1, 6)
    print(f"Roll {i+1}: {roll}")

    if roll == 6:
        count_6 += 1
        if previous_roll == 6:
            consecutive_6s += 1
    if roll == 1:
        count_1 += 1

    previous_roll = roll

print("\n--- Statistics ---")
print("Total rolls:", rolls)
print("Number of times 6 was rolled:", count_6)
print("Number of times 1 was rolled:", count_1)
print("Number of times two 6s were rolled in a row:", consecutive_6s)
```



```
C:\Users\Lenovo\Desktop\Shadowfox\Beginner\5>python 1.py
Roll 1: 3
Roll 2: 3
Roll 3: 2
Roll 4: 2
Roll 5: 6
Roll 6: 3
Roll 7: 1
Roll 8: 3
Roll 9: 5
Roll 10: 4
Roll 11: 3
Roll 12: 5
Roll 13: 5
Roll 14: 3
Roll 15: 3
Roll 16: 3
Roll 17: 2
Roll 18: 5
Roll 19: 2
Roll 20: 3

--- Statistics ---
Total rolls: 20
Number of times 6 was rolled: 1
Number of times 1 was rolled: 1
Number of times two 6s were rolled in a row: 0
```

2. Imagine you are doing a workout routine, and you have to complete 100 jumping jacks.

Write a program that:

Asks you to perform 10 jumping jacks at a time.

After each set, it asks, "Are you tired?"

If you reply "yes" or "y," it should ask if you want to skip the remaining sets.

If you reply "yes" or "y," it should break and print, "You completed a total of

jumping jacks."

For example, if you did only 30 jumping jacks and answered "yes," the program

will break and print, "You completed a total of 30 jumping jacks."

If you reply "no" or "n," it should continue and display how many jumping jacks

are remaining. After that, ask you again, "Are you tired?"

For example, if you answered "no," it should display that 70 jumping jacks are

remaining and ask you again, "Are you tired?"

If you reply "no" or "n," it should continue and display how many jumping jacks

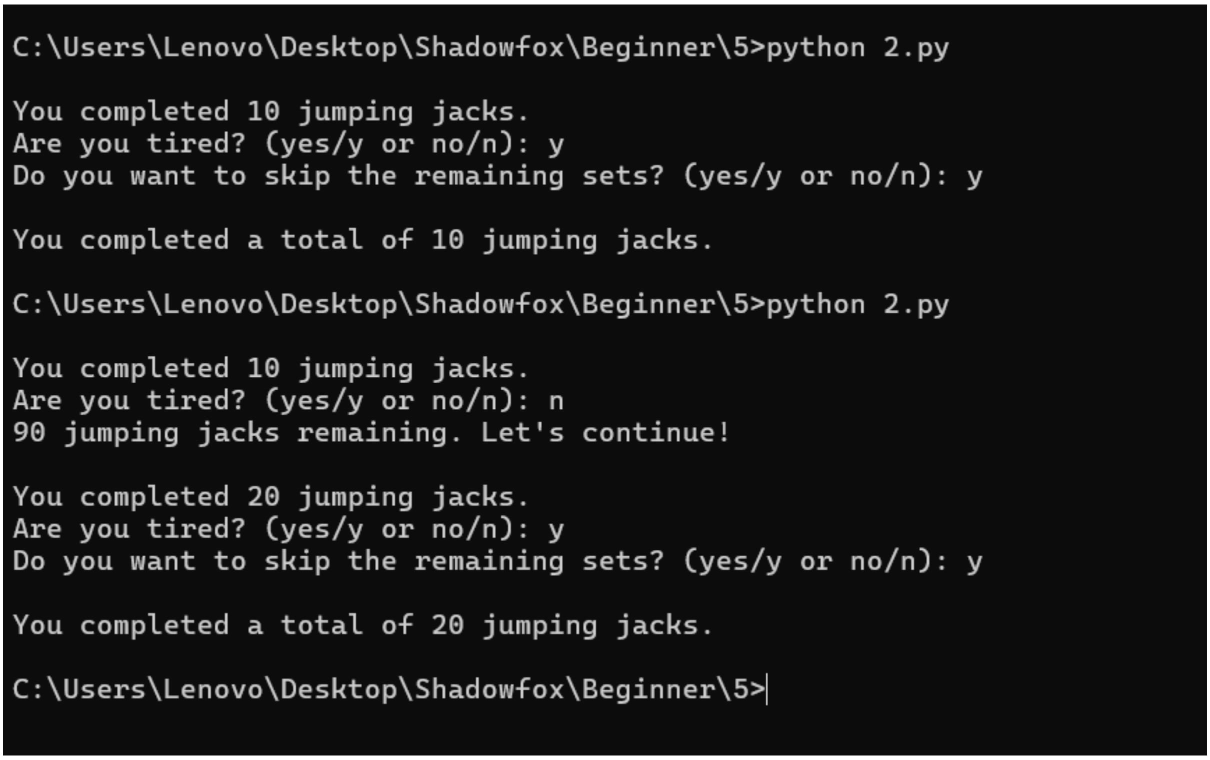
are remaining. After that, ask you again, "Are you tired?"

For example, if you answered "no," it should display that 70 jumping jacks are

remaining and ask you again, "Are you tired?"

If you complete all 100 jumping jacks, it should print, "Congratulations! You completed the workout," and stop the program

```
total = 100
set_size = 10
completed = 0

for i in range(0, total, set_size):
    completed += set_size
    print(f"\nYou completed {completed} jumping jacks.")

    if completed == total:
        print("🎉 Congratulations! You completed the workout.")
        break

    tired = input("Are you tired? (yes/y or no/n): ").strip().lower()

    if tired in ['yes', 'y']:
        skip = input("Do you want to skip the remaining sets? (yes/y or no/n): ").strip().lower()
        if skip in ['yes', 'y']:
            print(f"\nYou completed a total of {completed} jumping jacks.")
            break
        else:
            remaining = total - completed
            print(f"{remaining} jumping jacks remaining. Let's continue!")
    else:
        remaining = total - completed
        print(f"{remaining} jumping jacks remaining. Let's continue!")
```



```
C:\Users\Lenovo\Desktop\Shadowfox\Beginner\5>python 2.py

You completed 10 jumping jacks.
Are you tired? (yes/y or no/n): y
Do you want to skip the remaining sets? (yes/y or no/n): y

You completed a total of 10 jumping jacks.

C:\Users\Lenovo\Desktop\Shadowfox\Beginner\5>python 2.py

You completed 10 jumping jacks.
Are you tired? (yes/y or no/n): n
90 jumping jacks remaining. Let's continue!

You completed 20 jumping jacks.
Are you tired? (yes/y or no/n): y
Do you want to skip the remaining sets? (yes/y or no/n): y

You completed a total of 20 jumping jacks.

C:\Users\Lenovo\Desktop\Shadowfox\Beginner\5>
```