Name: Akhilesh Singh **Roll No: 186**

Lab. Assignment 6

PYTHON

Write Python Programs to implement conditional statements.

Sample Programs -

1. Write a Python program that checks if a given year is a leap year. Use conditional statements to implement the logic. **Code:**

```
def is_leap_year(year):
       if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
        return True
    else:
        return False
year = int(input("Enter a year: "))
if is_leap_year(year):
    print(f"{year} is a leap year.")
else:
    print(f"{year} is not a leap year.")
```

Output:

```
Enter a year: 2024
2024 is a leap year.
```

```
Enter a year: 2018
2018 is not a leap year.
```

1. Implement a program that takes an integer input from the user and prints all the prime numbers less than that number. Use loops and conditionals for this task.

Code:

```
def is_prime(n):
    if n <= 1:
        return False
    for i in range(2, int(n ** 0.5) + 1):
        if n % i == 0:
            return False
    return True

def print_primes_less_than(n):
    for num in range(2, n):
        if is_prime(num):
            print(num, end=" ")
```

```
number = int(input("Enter an integer: "))
print(f"Prime numbers less than {number} are:")
print_primes_less_than(number)
```

Output:

```
Enter an integer: 45
Prime numbers less than 45 are:
2 3 5 7 11 13 17 19 23 29 31 37 41 43
```

NUMPY Programs

1. **Creating NumPy Arrays Code:**

```
import numpy as np

arr1 = np.array([1, 2, 3, 4, 5])

print("1D Array:", arr1)

arr2 = np.array([[1, 2, 3], [4, 5, 6]])

print("2D Array:\n", arr2)
```

Output:

```
1D Array: [1 2 3 4 5]
2D Array:
 [[1 2 3]
 [4 5 6]]
```

2. **Accessing Elements in NumPy Array Code:**

```
arr2 = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
print("Element at (0, 0):", arr2[0, 0])
print("Element at (2, 1):", arr2[2, 1])
print("Element at (1, 2):", arr2[1, 2])
```

Output:

```
Element at (0, 0): 1
Element at (2, 1): 8
Element at (1, 2): 6
```

3. **Array Indexing and Slicing**

Code:

A) Indexing:

```
arr = [10, 20, 30, 40, 50]

print(arr[0])
print(arr[1])
print(arr[-1])
print(arr[-2])
```

Output:

```
10
20
50
40
```

B) Slicing:

```
arr = [10, 20, 30, 40, 50]
print(arr[1:4])
print(arr[:3])
print(arr[2:])
print(arr[-3:])
print(arr[::2])
```

Output:

```
[20, 30, 40]
[10, 20, 30]
[30, 40, 50]
[30, 40, 50]
[10, 30, 50]
```

4. **Array Operations (Addition, Multiplication, etc.) Code:**

```
import numpy as np

arr1 = np.array([10, 20, 30, 40])
arr2 = np.array([5, 10, 15, 20])
```

```
addition = arr1 + arr2
print("Addition:", addition)

subtraction = arr1 - arr2
print("Subtraction:", subtraction)

multiplication = arr1 * arr2
print("Multiplication:", multiplication)

division = arr1 / arr2
print("Division:", division)
```

Output:

```
Addition: [15 30 45 60]
Subtraction: [ 5 10 15 20]
Multiplication: [ 50 200 450 800]
Division: [2. 2. 2. 2.]
```

5. **Reshaping an Array Code:**

```
import numpy as np

arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])

reshaped_arr = arr.reshape(4, 3)
print(reshaped_arr)
```

Output:

```
[[ 1  2  3]
 [ 4  5  6]
 [ 7  8  9]
 [10 11 12]]
```

6. **Basic Statistical Operations (Mean, Median, Std Dev) Code:**

```
import numpy as np

data = np.array([10, 20, 30, 40, 50, 60, 70, 80, 90, 100])
mean_value = np.mean(data)
print(f"Mean: {mean_value}")
```

```
median_value = np.median(data)
print(f"Median: {median_value}")
std_dev_value = np.std(data)
print(f"Standard Deviation: {std_dev_value}")
```

Output:

```
Mean: 55.0
Median: 55.0
Standard Deviation: 28.722813232690143
```

7. **Creating Arrays with Random Numbers**

Code:

```
import numpy as np

random_array_integers = np.random.randint(1, 101, size=10)
print("1D Array of Random Integers:", random_array_integers)

random_array_floats = np.random.rand(10)
print("1D Array of Random Floats:", random_array_floats)

random_2d_array_integers = np.random.randint(1, 101, size=(3, 4))
print("2D Array of Random Integers:\n", random_2d_array_integers)

random_2d_array_floats = np.random.rand(3, 4)
print("2D Array of Random Floats:\n", random_2d_array_floats)

random_array_floats_range = np.random.uniform(5, 10, size=10)
print("1D Array of Random Floats in Range [5, 10]:",
random_array_floats_range)
```

Output:

```
1D Array of Random Integers: [17 58 75 93 63 55  8 42 85  7]
1D Array of Random Floats: [0.38939183 0.2190785  0.95168917 0.12753911 0.76994457 0.6083629
 0.62497751 0.71870893 0.05997279 0.21163794]
2D Array of Random Integers:
 [[24 29 62 77]
 [70 32 49 90]
 [60 83 58  4]]
2D Array of Random Floats:
 [[0.05753923 0.74129653 0.33276331 0.47762228]
 [0.84664026 0.51689847 0.73659407 0.20962394]
 [0.93891717 0.71093347 0.75900156 0.39228511]]
1D Array of Random Floats in Range [5, 10]: [5.92814875 7.06916645 9.19824976 9.04126224 7.54937531 9.57393584
 6.50856497 6.99495046 8.18387509 6.51060098]
```

8. **Concatenating and Splitting Arrays**

Code:

A) Concatenating Arrays

```
array1 = np.array([[1, 2, 3], [4, 5, 6]])

array2 = np.array([[7, 8, 9], [10, 11, 12]])

result = np.concatenate((array1, array2), axis=0)

print(result)
```

Output:

```
[[ 1  2  3]
 [ 4  5  6]
 [ 7  8  9]
 [10 11 12]]
```

B) Splitting Arrays

```
array2d = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])

result = np.split(array2d, 3, axis=0)

print(result)
```

Output:

Name: Ashik Shaikh **Roll No: 169**

```
[array([[1, 2, 3]]), array([[4, 5, 6]]), array([[7, 8, 9]])]
```