Raqib Ali AI-333134

Looping Structures

1. Write a Python program to print the numbers from 1 to 10 using a for loop.

```
for a in range (0,11):
    print(a)
```

```
0
1
2
3
4
5
6
7
8
9
10
```

2. Write a Python program to print the numbers from 20 to 1 using a while loop.

```
number=21

while number>1:
    number-=1
    print(number)
```

```
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
```

```
5
4
3
2
1
```

3. Write a program to print even numbers from 1 to 10.

```
number=0

while number<10:
    number+=2
    print(number)
```

```
2
4
6
8
10
```

4. Write a program that prompts the user to enter a number n and prints all the numbers from 1 to n.

```
number=0
n=int(input("Enter the number: "))

while number < n:
    number+=1
    print(number)
```

```
Enter the number: 21
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
```

```
20
21
```

5. Write a program that prompts the user to enter a number n, and then prints all the odd numbers between 1 and n.

```
number=-1
n=int(input("Enter the number: "))

while number <n:
    number+=2
    print(number)
```

```
Enter the number: 24
1
3
5
7
9
11
13
15
17
19
21
23
25
```

6. Write a program that prints 'Happy Birthday!' five times on screen.

```
for a in range(5):
    print("Happy Birthday")
```

```
Happy Birthday
Happy Birthday
Happy Birthday
Happy Birthday
Happy Birthday
```

7. Write a program that takes a number n as input from the user and generates the first n terms of the series formed by squaring the natural numbers.

```
# Get input from the user
n = int(input("Enter the number of terms (n): "))

# Check if the input is valid
if n <= 0:
    print("Please enter a positive integer.")
```

```
else:
    print(f"The first {n} terms of the series are:")
    for i in range(1, n + 1):
        print(i**2, end=" ")  # Square the number and print it

Enter the number of terms (n): 5
The first 5 terms of the series are:
1 4 9 16 25
```

8. Write a program that prompts the user to input a number and prints its multiplication table.

```
n=int(input("Enter the number: "))

for i in range(1,11):
    print(i*n,end=" ")

Enter the number: 2
2 4 6 8 10 12 14 16 18 20
```

9. Write a Python program to print the first 8 terms of an arithmetic progression starting with 3 and having a common difference of 4.The program should output the following sequence: 3 7 11 15 19 23 27 31

```
n=-1

while n<31:
    n+=4
    print(n,end=",")

3,7,11,15,19,23,27,31,
```

10. Write a Python program to print the first 6 terms of a geometric sequence starting with 2 and having a common ratio of 3. The program should output the following sequence: 2 6 18 54 162 486

```
n=2

while n<486:
    n*=3
    print(n,end=",")

6,18,54,162,486,
```

11. Write a program that asks the user for a positive integer value. The program should calculate the sum of all the integers from 1 up to the number entered. For example, if the user enters 20, the loop will find the sum of 1, 2, 3, 4, ... 20.

```
n=int(input("Enter a number: "))

for i in range(0,n):
    i+=1
    print(i,end=".")
```

```
Enter a number: 50
1.2.3.4.5.6.7.8.9.10.11.12.13.14.15.16.17.18.19.20.21.22.23.24.25.26.2
7.28.29.30.31.32.33.34.35.36.37.38.39.40.41.42.43.44.45.46.47.48.49.50
.
```

12. write a program that takes a positive integer N as input and calculates the sum of the reciprocals of all numbers from 1 up to N. The program should display the final sum.

```
def sum_of_reciprocals():
    try:
        # Take input from the user
        n = int(input("Enter a positive integer (N): "))

        # Validate the input
        if n <= 0:
            print("Please enter a positive integer greater than 0.")
            return

        # Calculate the sum of reciprocals
        reciprocal_sum = sum(1 / i for i in range(1, n + 1))

        # Display the result
        print(f"The sum of reciprocals from 1 to {n} is:
{reciprocal_sum}")

    except ValueError:
        print("Invalid input. Please enter a valid positive integer.")

# Call the function to execute the program
sum_of_reciprocals()
```

```
Enter a positive integer (N): 5
The sum of reciprocals from 1 to 5 is: 2.283333333333333
```

13. Write a program that prompts the user to enter a number and repeats this process 5 times. The program should accumulate the numbers entered and then display the final running total.

```
lst=[]
for a in range(5):
    n=int(input("Enter a number."))
    lst.append(n)
print(lst,lst[0]+lst[1]+lst[2]+lst[3]+lst[4])
```

```
Enter a number.15
Enter a number.10
Enter a number.12
Enter a number.15
Enter a number.14
[15, 10, 12, 15, 14] 66
```

14. Write a program that prompts the user to enter a positive integer and calculates its factorial. The factorial of a positive integer 'n' is denoted as 'n!' and is calculated by multiplying all the integers from 1 to 'n' together. For example, the factorial of 5 (denoted as 5!) is calculated as 1 x 2 x 3 x 4 x 5.

The program should display the factorial value if the input is a positive number, or display a message stating that the factorial does not exist for negative numbers. Additionally, for an input of zero, the program should output that the factorial of 0 is 1.

```
def calculate_factorial():
    try:
        # Prompt the user to enter a positive integer
        num = int(input("Enter a positive integer: "))

        # Handle the case for negative numbers
        if num < 0:
            print("Factorial does not exist for negative numbers.")
        # Handle the case for zero
        elif num == 0:
            print("The factorial of 0 is 1.")
        else:
            # Calculate the factorial using a loop
            factorial = 1
            for i in range(1, num + 1):
                factorial *= i

            # Display the result
```

```
            print(f"The factorial of {num} is {factorial}.")

    except ValueError:
        print("Invalid input. Please enter a valid positive integer.")

# Call the function to execute the program
calculate_factorial()
```

```
Enter a positive integer: 5
The factorial of 5 is 120.
```

15. Write a Python program that prompts the user to enter a base number and an exponent, and then calculates the power of the base to the exponent. The program should not use the exponentiation operator (**) or the math.pow() function. The program should handle both positive and negative exponents.

```
def calculate_power():
    try:
        # Prompt the user to enter the base and the exponent
        base = float(input("Enter the base number: "))
        exponent = int(input("Enter the exponent: "))

        # Initialize result to 1
        result = 1

        # Calculate power for positive exponent
        if exponent > 0:
            for _ in range(exponent):
                result *= base

        # Calculate power for negative exponent
        elif exponent < 0:
            for _ in range(-exponent):
                result *= base
            result = 1 / result

        # The result remains 1 if the exponent is 0
        # (since any number raised to the power of 0 is 1)

        # Display the result
        print(f"{base} raised to the power of {exponent} is {result}")

    except ValueError:
        print("Invalid input. Please enter numerical values for base
and integer values for exponent.")

# Call the function to execute the program
calculate_power()
```

```
Enter the base number: 2
Enter the exponent: 2
2.0 raised to the power of 2 is 4.0
```