LEVEL WISE QUESTIONS(Sample3)

1.Write a PYTHON program to produce following design (If user enters n value as 5)

A B C D E

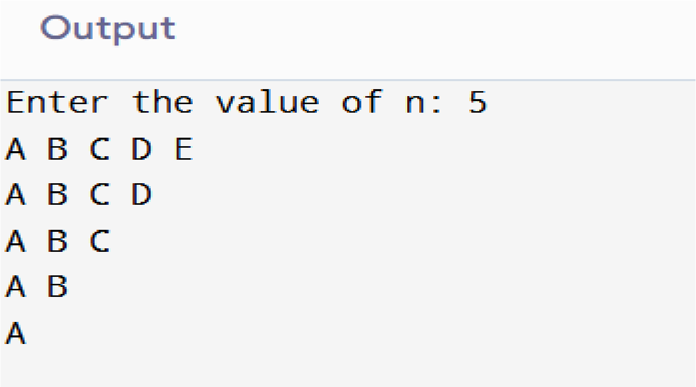
A B C D

A B C

A B

A

```
# Program to print the pattern
# A B C D E
# A B C D
# A B C
# A B
# A

def print_pattern(n):
    for i in range(n, 0, -1):
        for j in range(i):
            print(chr(65 + j), end=' ')
        print()

# Taking user input
n = int(input("Enter the value of n: "))
print_pattern(n)
```

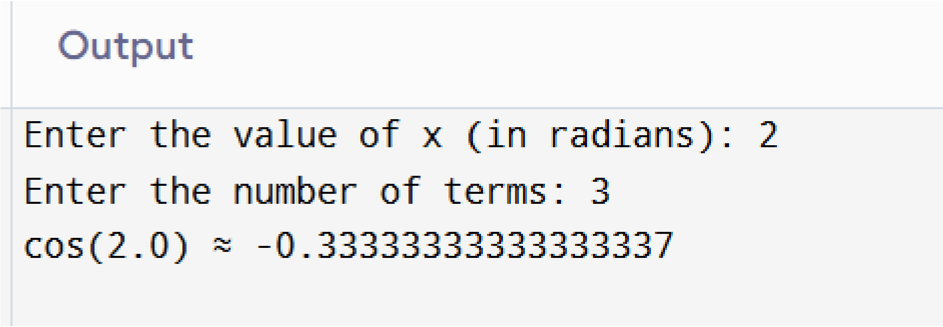


```
Enter the value of n: 5
A B C D E
A B C D
A B C
A B
A
```

2.Write a PYTHON program to compute the cosine series: cos(x) = 1 – x2 / 2! + x4 / 4! – x6 /

6! + … xn / n!

```
import math


def cosine_series(x, n_terms):

    cos_x = 0

    for k in range(n_terms):

        term = ((-1)**k * (x**(2*k))) / math.factorial(2*k)

        cos_x += term

    return cos_x


# User input

x = float(input("Enter the value of x (in radians): "))

n = int(input("Enter the number of terms: "))


# Calculate and display the result

result = cosine_series(x, n)

print(f"cos({x}) ≈ {result}")
```

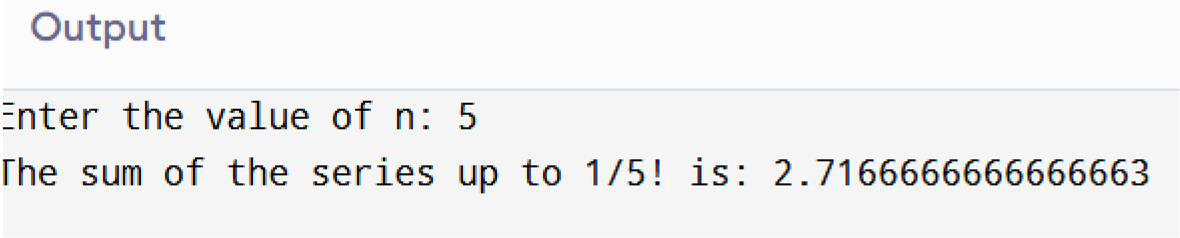


Output

```
Enter the value of x (in radians): 2
Enter the number of terms: 3
cos(2.0) ≈ -0.33333333333333337
```

3.Write a PYTHON program to sum the given sequence 1 + 1/ 1! + 1/ 2! + 1/3! + …. + 1/n!

```
import math


def compute_series(n):
    total = 0
    for i in range(n + 1):
        total += 1 / math.factorial(i)
    return total

# User input
n = int(input("Enter the value of n: "))

# Calculate and display the result
result = compute_series(n)
print(f"The sum of the series up to 1/{n}! is: {result}")
```

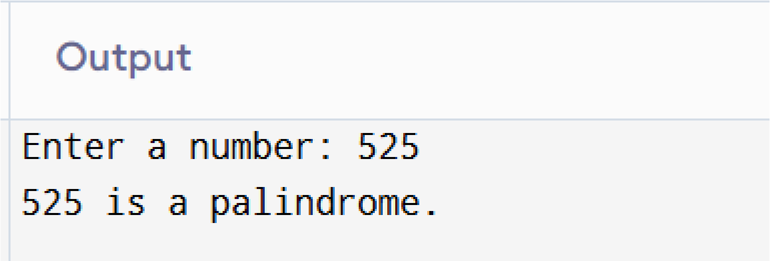


Output

```
Enter the value of n: 5
The sum of the series up to 1/5! is: 2.7166666666666663
```

4.Write a PYTHON program to check the entered number is palindrome or not

```
def is_palindrome(number):
    original = str(number)
    reversed_num = original[::-1]
    return original == reversed_num

# User input
num = input("Enter a number: ")

# Check and display result
if is_palindrome(num):
    print(f"{num} is a palindrome.")
else:
    print(f"{num} is not a palindrome.")
```

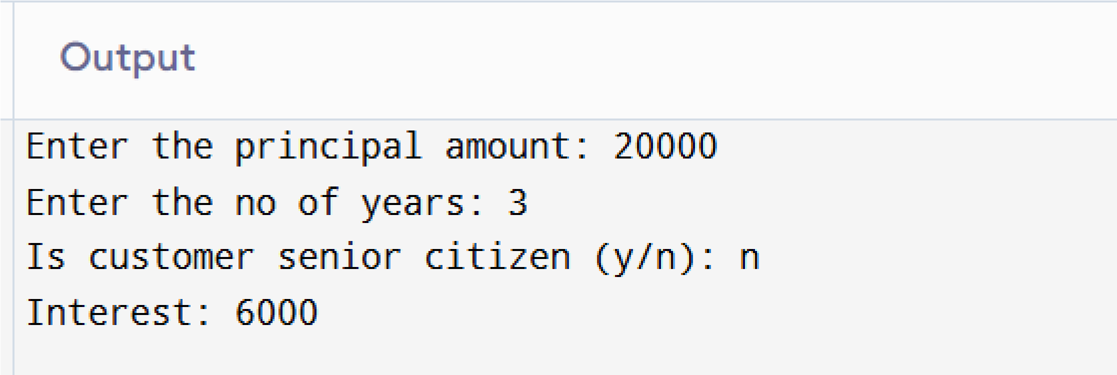


```
Enter a number: 525
525 is a palindrome.
```

5.Write a python program using function to calculate the simple interest. Suppose the customer is a senior citizen. He is being offered 12 percentage rate of interest; for all other

customers, the ROI is 10 percentage.

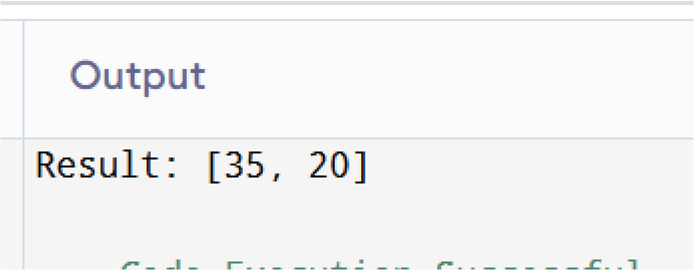
```
def calculate_simple_interest(principal, years, is_senior):
    if is_senior.lower() == 'y':
        rate = 12
    else:
        rate = 10
    interest = (principal * rate * years) / 100
    return interest

# Input from user
principal = float(input("Enter the principal amount: "))
years = int(input("Enter the no of years: "))
senior = input("Is customer senior citizen (y/n): ")

# Calculate interest
interest = calculate_simple_interest(principal, years, senior)

# Output
print(f"Interest: {int(interest)}")
```



```
Enter the principal amount: 20000
Enter the no of years: 3
Is customer senior citizen (y/n): n
Interest: 6000
```

6.Write a Python function sumsquare(l) that takes a nonempty list of integers and returns alist [odd,even], where odd is the sum of squares of all the odd numbers in l and even is thesum of squares of all the even numbers in l.

```
def sumsquare(l):
    odd_sum = 0
    even_sum = 0
    for num in l:
        if num % 2 == 0:
            even_sum += num ** 2
        else:
            odd_sum += num ** 2
    return [odd_sum, even_sum]


# Example usage:
numbers = [1, 2, 3, 4, 5]
result = sumsquare(numbers)
print("Result:", result)
```

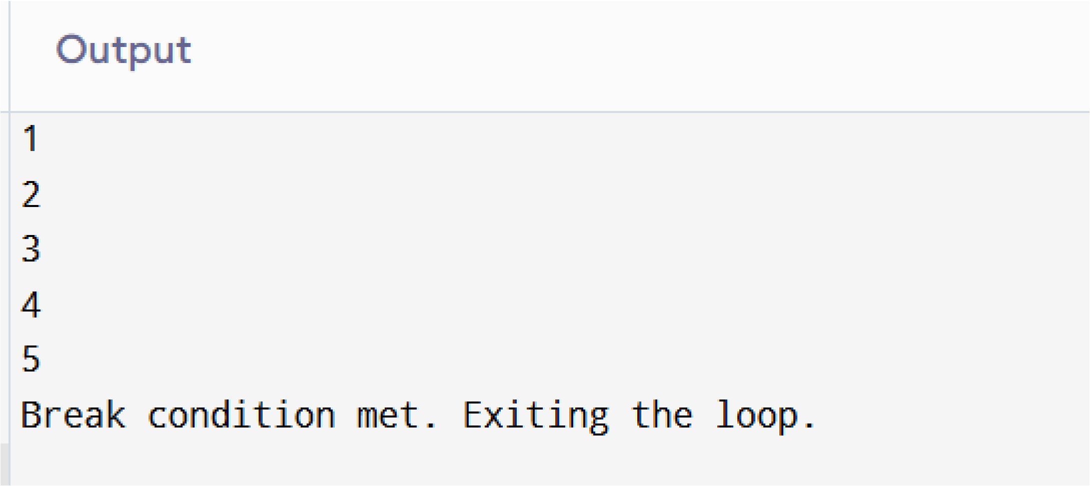


```
Result: [35, 20]

    Code Execution Successful
```

7.Write a PYTHON program to Print numbers using a loop with a break condition

```
# Program to print numbers from 1 to 10, but stop if number is 6
```

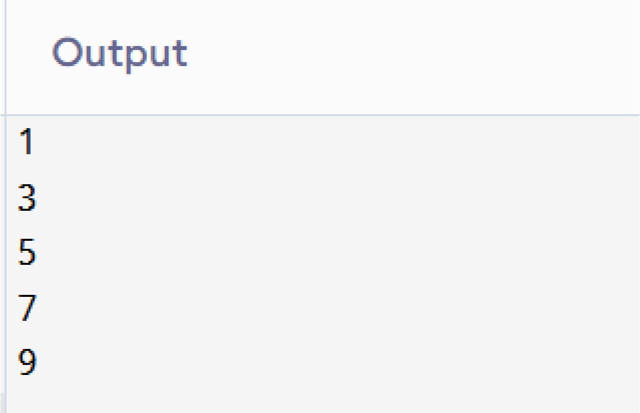
```
for i in range(1, 11):
    if i == 6:
        print("Break condition met. Exiting the loop.")
        break
    print(i)
```



```
1
2
3
4
5
Break condition met. Exiting the loop.
```

8.Write a PYTHON program to Skip even numbers using continue statement

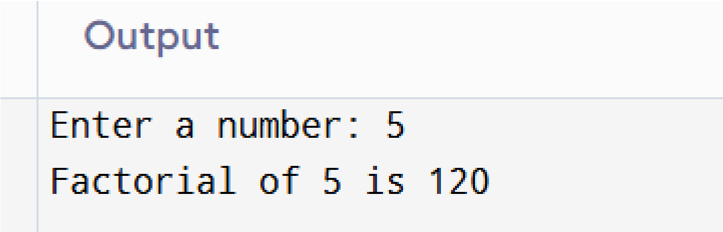
```
# Program to print only odd numbers from 1 to 10
for i in range(1, 11):
    if i % 2 == 0:
        continue  # Skip the rest of the loop for even numbers
    print(i)
```



```
1
3
5
7
9
```

9.Write a PYTHON program to Find factorial of a number

```
def factorial(n):
    result = 1
    for i in range(2, n + 1):
        result *= i
    return result

# User input
num = int(input("Enter a number: "))

# Check for negative input
if num < 0:
    print("Factorial is not defined for negative numbers.")
else:
    print(f"Factorial of {num} is {factorial(num)}")
```

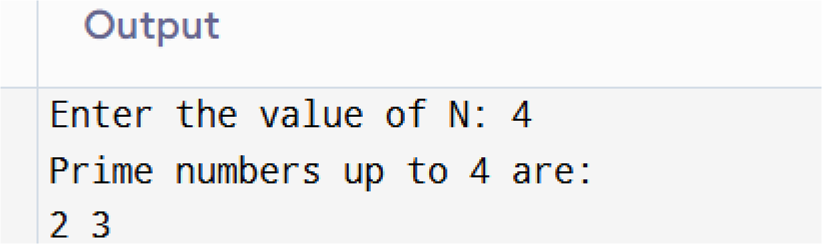


```
Enter a number: 5
Factorial of 5 is 120
```

10.Write a PYTHON program to Find prime numbers up to N

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

# User input
N = int(input("Enter the value of N: "))

# Find and print prime numbers up to N
```

```
print(f"Prime numbers up to {N} are:")
for i in range(2, N + 1):
    if is_prime(i):
        print(i, end=' ')
```

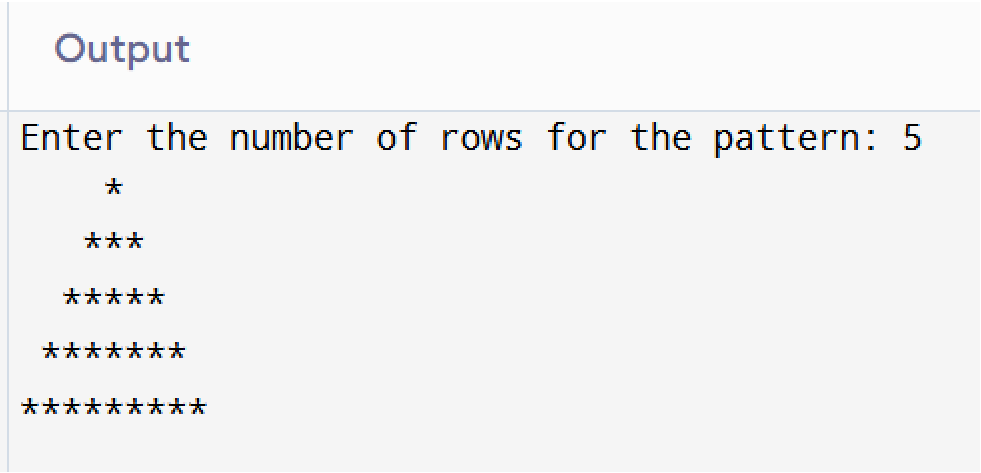


Output

```
Enter the value of N: 4
Prime numbers up to 4 are:
2 3
```

11.Write a PYTHON program to Print a pattern using nested loops

```
def print_pattern(n):
    for i in range(1, n + 1):
        # Print leading spaces
        for j in range(n - i):
            print(" ", end="")
        # Print stars
        for k in range(2 * i - 1):
            print("*", end="")
        # Move to the next line
        print()

# User input
n = int(input("Enter the number of rows for the pattern: "))

# Print the pattern
print_pattern(n)
```



```
Enter the number of rows for the pattern: 5
    *
   ***
  *****
 *******
*********
```