**Question 1:**

**Write a program using generator to print numbers divisible by both 5 and 7 between 0 and n in comma-separated form:**

def divisible\_by\_5\_and\_7(n):

for i in range(0, n+1):

if i % 5 == 0 and i % 7 == 0:

yield i

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

result = [str(i) for i in divisible\_by\_5\_and\_7(n)]

print(",".join(result))

**Example Input:**

100

**Output:**

0,35,70

**Question 2:**

**Write a program using generator to print the even numbers between 0 and n in comma-separated form:**

def even\_numbers(n):

for i in range(0, n+1, 2):

yield i

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

result = [str(i) for i in even\_numbers(n)]

print(",".join(result))

**Example Input:**

10

**Output:**

0,2,4,6,8,10

**Question 3:**

**The Fibonacci Sequence is computed based on the following formula:**

* f(n) = 0 if n = 0
* f(n) = 1 if n = 1
* f(n) = f(n-1) + f(n-2) if n > 1

Write a program using list comprehension to print the Fibonacci Sequence for a given n:

def fibonacci(n):

fib\_sequence = [0, 1] + [0] \* (n-1)

[fib\_sequence.\_\_setitem\_\_(i, fib\_sequence[i-1] + fib\_sequence[i-2]) for i in range(2, n+1)]

return fib\_sequence[:n+1]

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

result = [str(num) for num in fibonacci(n)]

print(",".join(result))

**Example Input:**

7

**Output:**

0,1,1,2,3,5,8,13

**Question 4:**

**Write a program to print the username from an email address in the format "**[**username@companyname.com**](mailto:username@companyname.com)**":**

def get\_username(email):

return email.split('@')[0]

email = input("Enter the email address: ")

print(get\_username(email))

**Example Input:**

john@google.com

**Output:**

john

**Question 5:**

**Define a class named Shape and its subclass Square. The Square class has an \_\_init\_\_ function that takes the length as an argument. Both classes have an area function that can print the area of the shape where Shape's area is 0 by default.**

class Shape:

def area(self):

return 0

class Square(Shape):

def \_\_init\_\_(self, length):

self.length = length

def area(self):

return self.length \*\* 2

# Example usage

shape = Shape()

square = Square(5)

print(f"Area of Shape: {shape.area()}")

print(f"Area of Square: {square.area()}")

**Output:**

Area of Shape: 0

Area of Square: 25