Question 1:

Please write a program using generator to print the numbers which can be divisible by 5 and 7 between 0 and n in comma separated form while n is input by console.

Example:  
If the following n is given as input to the program:

100

Then, the output of the program should be:

0,35,70

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

for num in range(n + 1):

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

yield num

# Test the program

if \_\_name\_\_ == "\_\_main\_\_":

try:

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

if n < 0:

print("n should be a non-negative integer.")

else:

result = divisible\_by\_5\_and\_7(n)

print("Output:", ",".join(map(str, result)))

except ValueError:

print("Invalid input! Please enter a non-negative integer value for n.")

Example usage:

Enter the value of n: 100

Output: 0,35,70

Question 2:

Please write a program using generator to print the even numbers between 0 and n in comma separated form while n is input by console.

Example:  
If the following n is given as input to the program:

10

Then, the output of the program should be:

0,2,4,6,8,10

ANS: def even\_numbers\_generator(n):

for num in range(n + 1):

if num % 2 == 0:

yield num

# Test the program

if \_\_name\_\_ == "\_\_main\_\_":

try:

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

if n < 0:

print("n should be a non-negative integer.")

else:

result = even\_numbers\_generator(n)

print("Output:", ",".join(map(str, result)))

except ValueError:

print("Invalid input! Please enter a non-negative integer value for n.")

Example usage:

Enter the value of n: 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

Please write a program using list comprehension to print the Fibonacci Sequence in comma separated form with a given n input by console.

Example:  
If the following n is given as input to the program:

7

Then, the output of the program should be:

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

ANS: def fibonacci\_sequence(n):

fib\_sequence = [0, 1]

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

return fib\_sequence[:n]

# Test the program

if \_\_name\_\_ == "\_\_main\_\_":

try:

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

if n < 0:

print("n should be a non-negative integer.")

else:

result = fibonacci\_sequence(n)

print("Output:", ",".join(map(str, result)))

except ValueError:

print("Invalid input! Please enter a non-negative integer value for n.")

Example usage:

Enter the value of n: 7

Output: 0,1,1,2,3,5,8

Question 4:

Assuming that we have some email addresses in the "[username@companyname.com](mailto:username@companyname.com)" format, please write program to print the user name of a given email address. Both user names and company names are composed of letters only.

Example:  
If the following email address is given as input to the program:

[john@google.com](mailto:john@google.com)

Then, the output of the program should be:

John

ANS: def extract\_username(email):

try:

username = email.split('@')[0]

if username.isalpha():

return username

else:

return None

except IndexError:

return None

# Test the program

if \_\_name\_\_ == "\_\_main\_\_":

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

username = extract\_username(input\_email)

if username:

print("Username:", username)

else:

print("Invalid email address or username format.")

Example Usage:

Enter the email address: john@google.com

Username: john

Question 5:

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

ANS: class Shape:

def area(self):

return 0

class Square(Shape):

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

self.length = length

def area(self):

return self.length \* self.length

# Test the classes

if \_\_name\_\_ == "\_\_main\_\_":

shape = Shape()

square = Square(5)

print("Area of the Shape:", shape.area()) # Output: 0

print("Area of the Square:", square.area()) # Output: 25

Example usage:

Area of the Shape: 0

Area of the Square: 25