ALIGARH MUSLIM UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

FACULTY OF SCIENCE

Computer Lab work

CAMS-3P01: Laboratory Course III

M.C.A 3Rd SEMESTER

SUBMITTED TO

MR. IMSHAD AHMAD KHAN

MR. MOHD FARAZ

MR. MD NAVED IQBAL

MR. TIPU SULTAN

DR. PARVEJ AALAM

MR. ZUBAIR AHMAD HASHMI

SUBMITTED BY

MOHD ASJAD

23CAMSA127

GK8330

WEEK 1

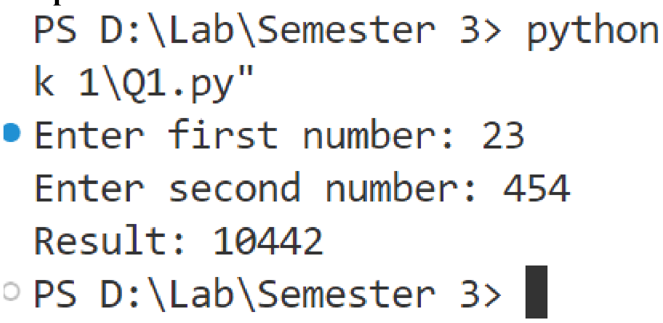
1. Write a program to find the product of two user-supplied integers and if the product is equal to or lower than 5000, then return the sum of the two numbers.

   Program:

```
def product_and_sum(a, b):
    product = a * b
    if product <= 5000:
        return a + b
    return product

num1 = int(input("Enter first number: "))
num2 = int(input("Enter second number: "))
result = product_and_sum(num1, num2)
print("Result:", result)
```

   Output:



```
PS D:\Lab\Semester 3> python
k 1\Q1.py"
Enter first number: 23
Enter second number: 454
Result: 10442
PS D:\Lab\Semester 3>
```

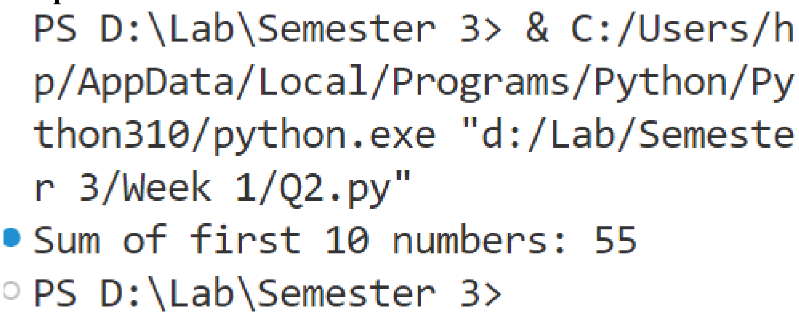
2. Write a program to print the sum of the first 10 numbers.

   Program:

```
def sum_of_first_10_numbers():
    return sum(range(1, 11))

print("Sum of first 10 numbers:", sum_of_first_10_numbers())
```

   Output:



```
PS D:\Lab\Semester 3> & C:/Users/h
p/AppData/Local/Programs/Python/Py
thon310/python.exe "d:/Lab/Semeste
r 3/Week 1/Q2.py"
Sum of first 10 numbers: 55
PS D:\Lab\Semester 3>
```

3. Write a program to iterate through a supplied list of 20 numbers and print only those numbers which are divisible by 5.

Program:

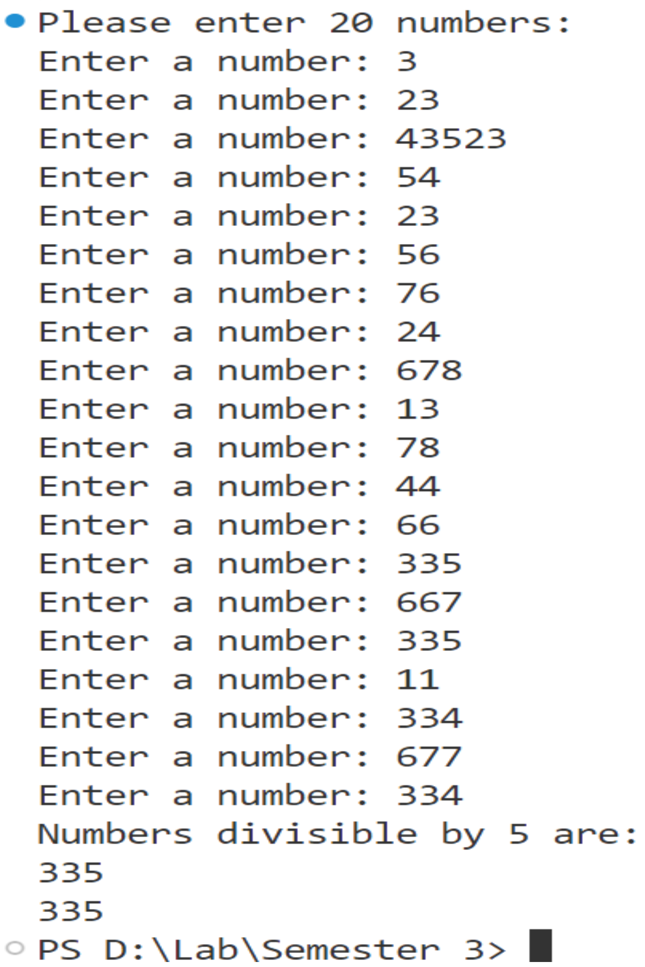
```
def print_divisible_by_5(numbers):
    for num in numbers:
        if num % 5 == 0:
            print(num)

numbers = []

print("Please enter 20 numbers:")
for _ in range(20):
    number = int(input("Enter a number: "))
    numbers.append(number)

print("Numbers divisible by 5 are:")
print_divisible_by_5(numbers)
```

Output:



```
● Please enter 20 numbers:
  Enter a number: 3
  Enter a number: 23
  Enter a number: 43523
  Enter a number: 54
  Enter a number: 23
  Enter a number: 56
  Enter a number: 76
  Enter a number: 24
  Enter a number: 678
  Enter a number: 13
  Enter a number: 78
  Enter a number: 44
  Enter a number: 66
  Enter a number: 335
  Enter a number: 667
  Enter a number: 335
  Enter a number: 11
  Enter a number: 334
  Enter a number: 677
  Enter a number: 334
  Numbers divisible by 5 are:
  335
  335
○ PS D:\Lab\Semester 3>
```

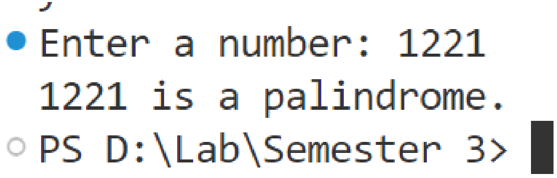
4. Write a program to check if the given number is a palindrome.

Program:

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

num = int(input("Enter a number: "))
if is_palindrome(num):
    print(f"{num} is a palindrome.")
else:
    print(f"{num} is not a palindrome.")
```

Output:



```
● Enter a number: 1221
  1221 is a palindrome.
○ PS D:\Lab\Semester 3>
```

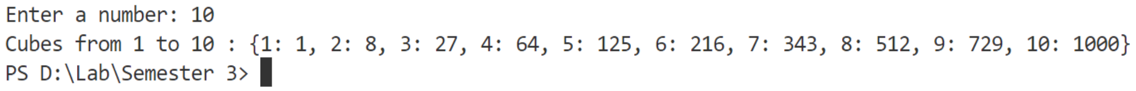
5. Write a program to calculate the cube of all numbers from 1 to a given number.

Program:

```
def cubes_up_to_n(n):
    return {i: i**3 for i in range(1, n+1)}

n = int(input("Enter a number: "))
print("Cubes from 1 to", n, ":", cubes_up_to_n(n))
```

Output:



```
Enter a number: 10
Cubes from 1 to 10 : {1: 1, 2: 8, 3: 27, 4: 64, 5: 125, 6: 216, 7: 343, 8: 512, 9: 729, 10: 1000}
PS D:\Lab\Semester 3>
```

WEEK 2

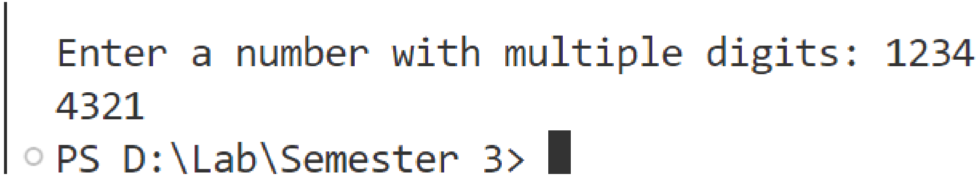
1. Write a program to extract each digit from an integer in reverse order.

Program:

```
def extract_digits_reverse(num):
    while num > 0:
        digit = num % 10
        print(digit, end="")
        num //= 10

num = int(input("Enter a number with multiple digits: "))
extract_digits_reverse(num)
```

Output:



```
Enter a number with multiple digits: 1234
4321
○ PS D:\Lab\Semester 3> █
```

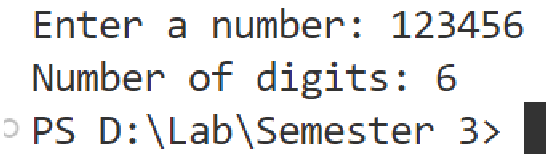
2. Write a program to count the total number of digits in a number using a while loop.

Program:

```
def count_digits(num):
    count = 0
    while num != 0:
        num //= 10
        count += 1
    return count

num = int(input("Enter a number: "))
print("Number of digits:", count_digits(num))
```

Output:



```
Enter a number: 123456
Number of digits: 6
○ PS D:\Lab\Semester 3> █
```

3. Write a program to display all prime numbers within a range.

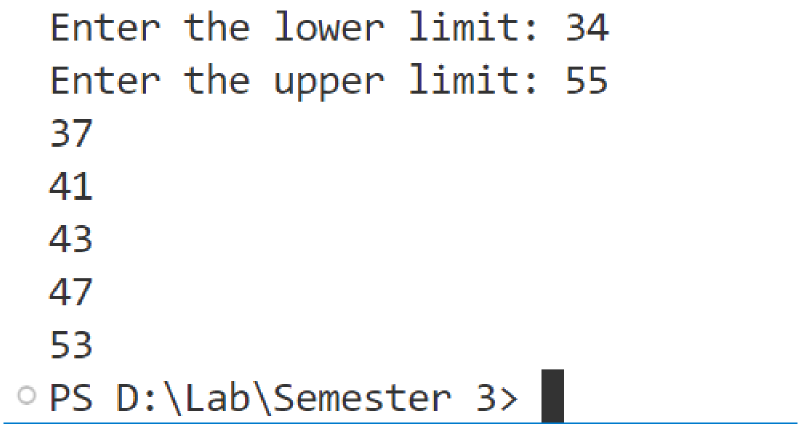
Program:

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

```
lower = int(input("Enter the lower limit: "))
upper = int(input("Enter the upper limit: "))

for num in range(lower, upper + 1):
   if is_prime(num):
      print(num)
```

Output:



```
Enter the lower limit: 34
Enter the upper limit: 55
37
41
43
47
53
PS D:\Lab\Semester 3>
```

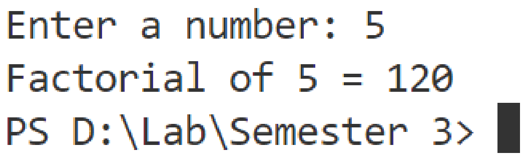
4. Write a program to use a loop to find the factorial of a given number.

Program:

```
def factorial(n):
   if n == 0:
      return 1
   fact = 1
   for i in range(1, n + 1):
      fact *= i
   return fact

num = int(input("Enter a number: "))
print(f"Factorial of {num} = {factorial(num)}")
```

Output:



```
Enter a number: 5
Factorial of 5 = 120
PS D:\Lab\Semester 3>
```

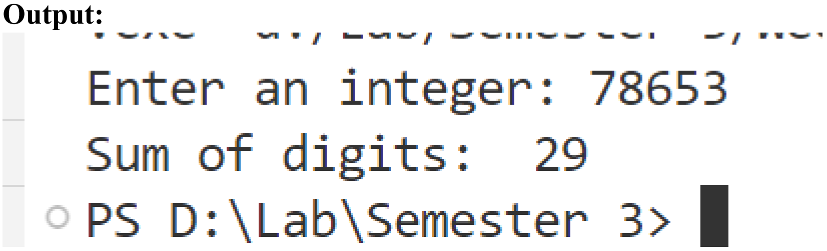
5. Write a program to find the sum of the digits of a supplied integer.

Program:

```
def sum_of_digits(num):
   sum =0
   while num > 0:
      digit = num % 10
      sum = sum +digit
      num  = num// 10
```

```
    return sum
num = int(input("Enter an integer: "))
print("Sum of digits: ", sum_of_digits(num))
```

Output:



```
Enter an integer: 78653
Sum of digits:  29
PS D:\Lab\Semester 3>
```

WEEK 3

1. Write a program to print the following pattern using the for loop:

5 4 3 2 1

4 3 2 1

3 2 1

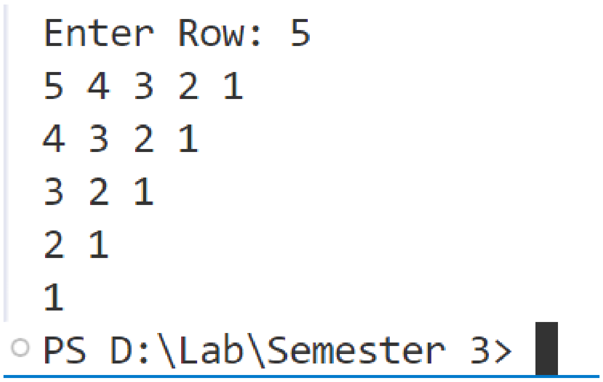
2 1

1

Program:

```
rows = int(input("Enter Row: "))
for i in range(rows, 0,-1):
    for j in range(i, 0,-1):
        print(j, end=" ")
    print()
```

Output:



```
Enter Row: 5
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1
PS D:\Lab\Semester 3>
```

2. Write a program to print the following star pattern using the for loop:

*

* *

* * *

* * * *

* * * * *

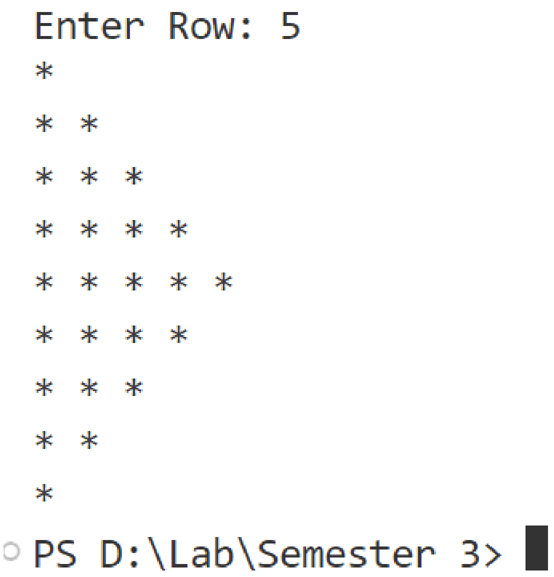
* * * *

```
* * *
* *
*
```

Program:

```
rows = int(input('Enter Row: '))
for i in range(1, rows + 1):
    for j in range(i):
        print("*", end=" ")
    print()
for i in range(rows - 1, 0, -1):
    for j in range(i):
        print("*", end=" ")
    print()
```

Output:



```
Enter Row: 5
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
PS D:\Lab\Semester 3>
```

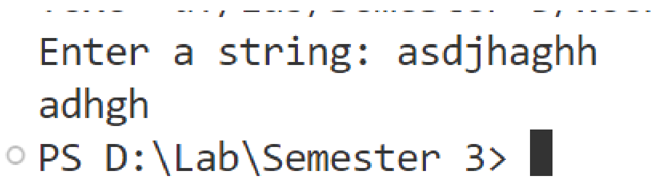
3. Write a program to print characters from a string which are present at even index numbers.

Program:

```
def print_even_index_characters(string):
    result = ""
    for i in range(len(string)):
        if i % 2 == 0:
            result += string[i]
    print(result)


input_string = input("Enter a string: ")
print_even_index_characters(input_string)
```

Output:



```
Enter a string: asdjhaghh
adhgh
PS D:\Lab\Semester 3>
```

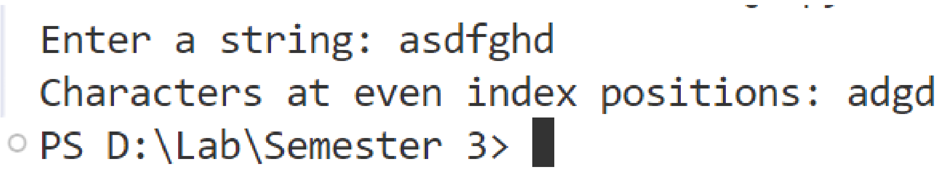
4. Write a program to accept a string from the user and display characters that are present at even index numbers.

Program:

```
def print_even_index_characters_from_user_input():
    string = input("Enter a string: ")
    result = ""
    for i in range(len(string)):
        if i % 2 == 0:
            result += string[i]
    print("Characters at even index positions:", result)


print_even_index_characters_from_user_input()
```

Output:



```
Enter a string: asdfghd
Characters at even index positions: adgd
PS D:\Lab\Semester 3>
```

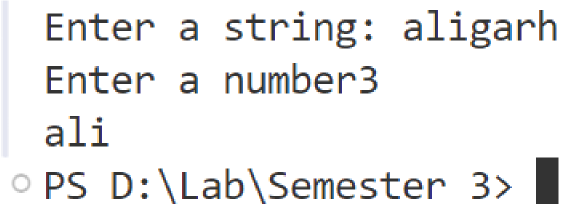
6. Write a program to remove characters from a string starting from the nth position to the last and return a new string. Example: remove_chars("aligarh", 3) should output ali.

Program:

```
def remove_chars(string, n):
    if n > len(string):
        return ""
    return string[:n]

str =  input("Enter a string: ")
num =  int(input("Enter a number"))
new_string = remove_chars(str, num)
print(new_string)
```

Output:



```
Enter a string: aligarh
Enter a number3
ali
PS D:\Lab\Semester 3>
```

Week 4

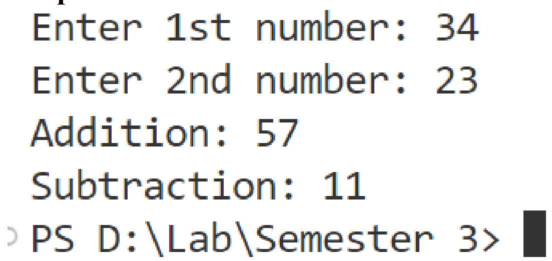
1. Write a program to create a function cal_sum_sub() that accepts two variables and calculates addition and subtraction. Also, it must return both addition and subtraction in a single return call.

   Program:

```
def cal_sum_sub(a, b):
    addition = a + b
    subtraction = a - b
    return addition, subtraction

a =  int(input("Enter 1st number: "))
b =  int(input("Enter 2nd number: "))
result = cal_sum_sub(a, b)
print("Addition:", result[0])
print("Subtraction:", result[1])
```

   Output:



```
Enter 1st number: 34
Enter 2nd number: 23
Addition: 57
Subtraction: 11
PS D:\Lab\Semester 3>
```

2. Write a function to return True if the first and last number of a given list are the same. If the numbers are different, return False.

   Program:

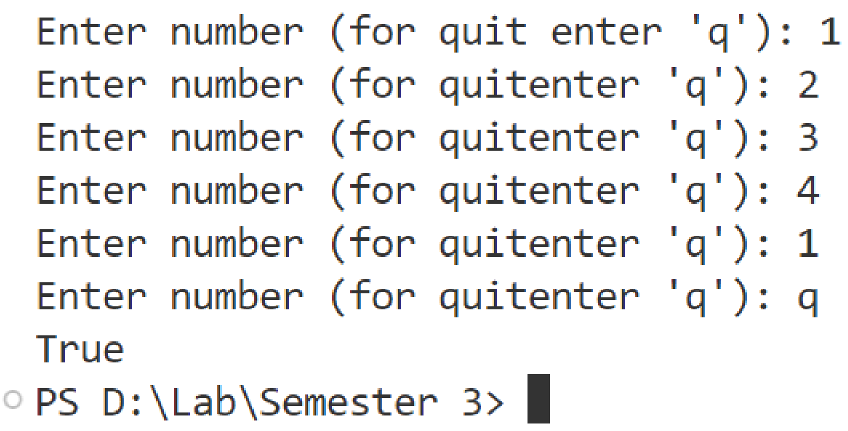
```
def is_first_last_same(numbers):
    return numbers[0] == numbers[-1]

numbers = []
a =  input("Enter number (for quit enter 'q'): ")
while a != 'q':
    try:
        numbers.append(int(a))
    except:
        print("enter valid input")

    a =  input("Enter number (for quitenter 'q'): ")


print(is_first_last_same(numbers))
```

   Output:



```
Enter number (for quit enter 'q'): 1
Enter number (for quitenter 'q'): 2
Enter number (for quitenter 'q'): 3
Enter number (for quitenter 'q'): 4
Enter number (for quitenter 'q'): 1
Enter number (for quitenter 'q'): q
True
○ PS D:\Lab\Semester 3> █
```

3. Given a list of numbers, write a program to turn every item of the list into its square.
   Program:

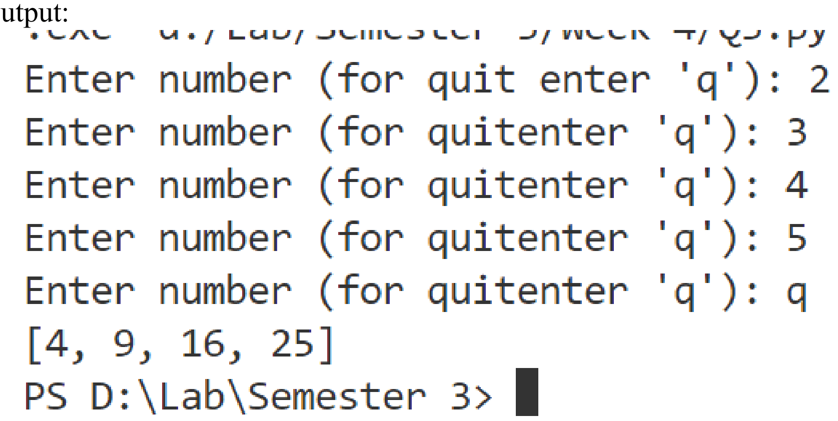
```
def square_list(lst):
    return [x**2 for x in lst]

numbers = []
a =  input("Enter number (for quit enter 'q'): ")
while a != 'q':
    try:
        numbers.append(int(a))
    except:
        print("enter valid input")

    a =  input("Enter number (for quitenter 'q'): ")

squared_numbers = square_list(numbers)
print(squared_numbers)
```

Output:



```
.exe  d:/Lab/Semester 3/Week 4/Q3.py
Enter number (for quit enter 'q'): 2
Enter number (for quitenter 'q'): 3
Enter number (for quitenter 'q'): 4
Enter number (for quitenter 'q'): 5
Enter number (for quitenter 'q'): q
[4, 9, 16, 25]
PS D:\Lab\Semester 3> █
```

4. Given two Python lists, write a program to iterate both lists simultaneously and display items from list 1 in original order and items from list 2 in reverse order.
   Program:

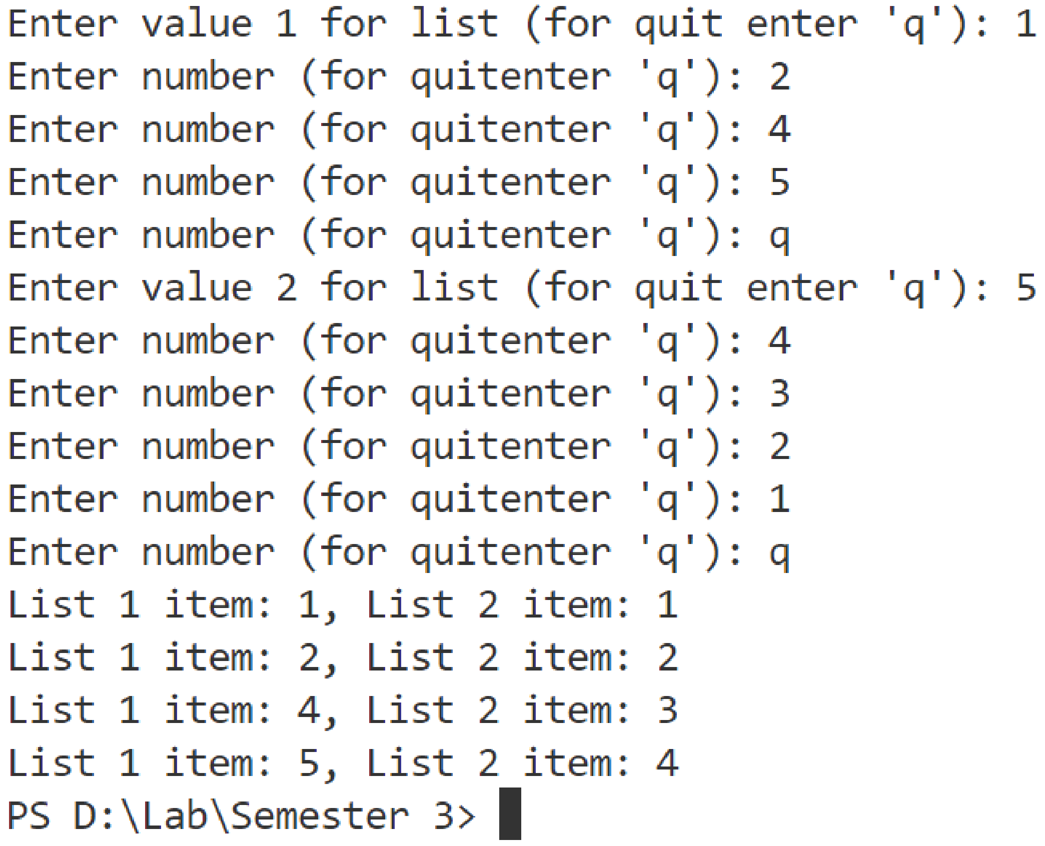
```
def iterate_lists(lst1, lst2):
    for x, y in zip(lst1, lst2[::-1]):
        print(f"List 1 item: {x}, List 2 item: {y}")

def input_fun(a):
    l =[]
    a =  input(f"Enter value {a} for list (for quit enter 'q'): ")
    while a != 'q':
        try:
            l.append(int(a))
        except:
            pass


        a =  input("Enter number (for quitenter 'q'): ")
    return l;


list1 = input_fun(1)
print("Enter value for 2nd list")
list2  =input_fun(2)
iterate_lists(list1, list2)
```

Output:



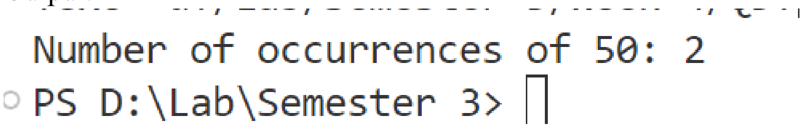
```
Enter value 1 for list (for quit enter 'q'): 1
Enter number (for quitenter 'q'): 2
Enter number (for quitenter 'q'): 4
Enter number (for quitenter 'q'): 5
Enter number (for quitenter 'q'): q
Enter value 2 for list (for quit enter 'q'): 5
Enter number (for quitenter 'q'): 4
Enter number (for quitenter 'q'): 3
Enter number (for quitenter 'q'): 2
Enter number (for quitenter 'q'): 1
Enter number (for quitenter 'q'): q
List 1 item: 1, List 2 item: 1
List 1 item: 2, List 2 item: 2
List 1 item: 4, List 2 item: 3
List 1 item: 5, List 2 item: 4
PS D:\Lab\Semester 3> 
```

5. Write a program to count the number of occurrences of item 50 in the tuple tp1 = (50, 10, 60, 70, 50).

   Program:

```
tp1 = (50, 10, 60, 70, 50)
count_50 = tp1.count(50)
print("Number of occurrences of 50:", count_50)
```

   Output:



```
Number of occurrences of 50: 2
PS D:\Lab\Semester 3> 
```

Week 5

1. Write a program to generate a 6-digit random secure OTP.

Program:

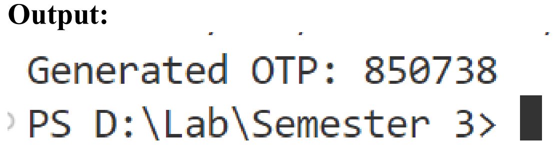
```
import random
import math

def generate_otp():
    digits = "0123456789"
    OTP = ""
    for i in range(6) :
        OTP += digits[math.floor(random.random() * 10)]

    return OTP

print("Generated OTP:", generate_otp())
```

Output:



```
Generated OTP: 850738
PS D:\Lab\Semester 3>
```

2. Write a program to pick a random character from a user-supplied string.

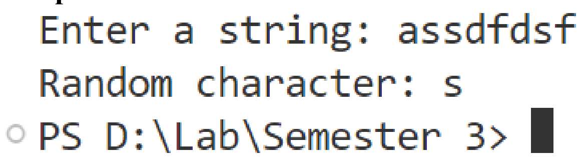
Program:

```
import random

def pick_random_char(user_string):
    return random.choice(user_string)

user_string = input("Enter a string: ")
print("Random character:", pick_random_char(user_string))
```

Output:



```
Enter a string: assdfdsfsds
Random character: s
PS D:\Lab\Semester 3>
```

3. Write a program to generate a random password that meets the following conditions:

a. Password length must be 10 characters long.

b. It must contain at least 2 uppercase letters, 1 digit, and 1 special symbol.

Program:

```
import random
import string
```

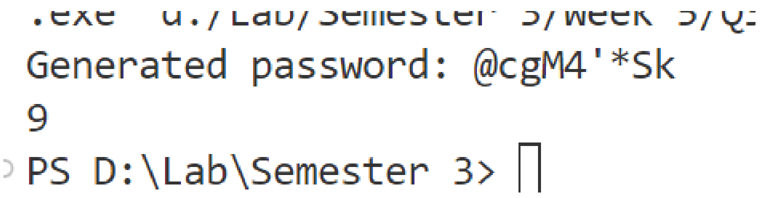
```
def generate_password():
    upper_case = random.choices(string.ascii_uppercase, k=2)
    digit = random.choice(string.digits)
    remaining_chars = random.choices(string.ascii_letters + string.digits + string.punctuation, k=6)

    password_list = upper_case + [digit] + remaining_chars
    random.shuffle(password_list)

    return ''.join(password_list)

print("Generated password:", generate_password())
print(len(generate_password()))
```

Output:



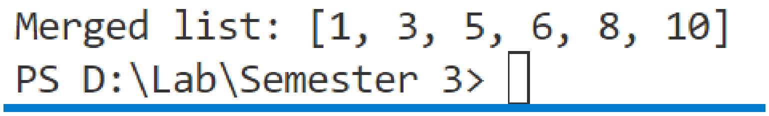
```
.exe  d:/Lab/Semester 3/Week 3/Q:
Generated password: @cgM4'*Sk
9
PS D:\Lab\Semester 3>
```

4. Given two lists of numbers, write a program to create a new list containing odd numbers from the first list and even numbers from the second list.
   Program:

```
def merge_odd_even(lst1, lst2):
    odd_numbers = [x for x in lst1 if x % 2 != 0]
    even_numbers = [x for x in lst2 if x % 2 == 0]
    return odd_numbers + even_numbers

# Example usage
list1 = [1, 2, 3, 4, 5]
list2 = [6, 7, 8, 9, 10]
result = merge_odd_even(list1, list2)
print("Merged list:", result)
```

Output:



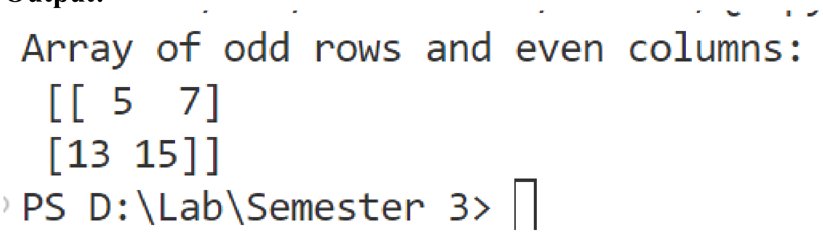
```
Merged list: [1, 3, 5, 6, 8, 10]
PS D:\Lab\Semester 3>
```

5. Write a program to create a numpy array and return an array of odd rows and even columns from the numpy array.
   Program:

```
import numpy as np
```

```
arr = np.array([[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12], [13, 14, 15, 16]])
result = arr[1::2, ::2]
print("Array of odd rows and even columns:\n", result)
```

Output:



```
Array of odd rows and even columns:
 [[ 5  7]
 [13 15]]
PS D:\Lab\Semester 3>
```

6. Write a program to create a numpy array and sort it as per the following cases:

a. Case 1: Sort the array by the second row.

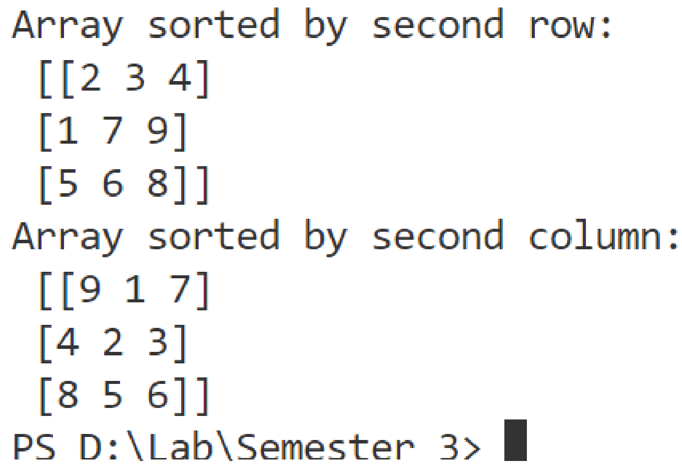
b. Case 2: Sort the array by the second column.

Program:

```
import numpy as np


# a) Sort array by second row
def sort_by_second_row(arr):
   return arr[:, arr[1, :].argsort()]
# b) Sort array by second column
def sort_by_second_column(arr):
   return arr[arr[:, 1].argsort()]
arr = np.array([[4, 2, 3], [9, 1, 7], [8, 5, 6]])
# Case 1
sorted_by_row = sort_by_second_row(arr)
print("Array sorted by second row:\n", sorted_by_row)
# Case 2
sorted_by_column = sort_by_second_column(arr)
print("Array sorted by second column:\n", sorted_by_column)
```

Output:



```
Array sorted by second row:
 [[2 3 4]
 [1 7 9]
 [5 6 8]]
Array sorted by second column:
 [[9 1 7]
 [4 2 3]
 [8 5 6]]
PS D:\Lab\Semester 3>
```