Atma Ram Sanatan Dharma College
University of Delhi

Programming in Python

Practical File for Paper Code 32343306

Submitted By
NITIN KUMAR
DWIVEDI
College Roll No. 21/18087
BSc (Hons) Computer Science

Submitted To
Mr Lokesh Kumar Shrivastav
Department of Computer Science

INDEX

8	Write a Python program to perform the given operations using lists.	--/09/2020

PRACTICAL 1

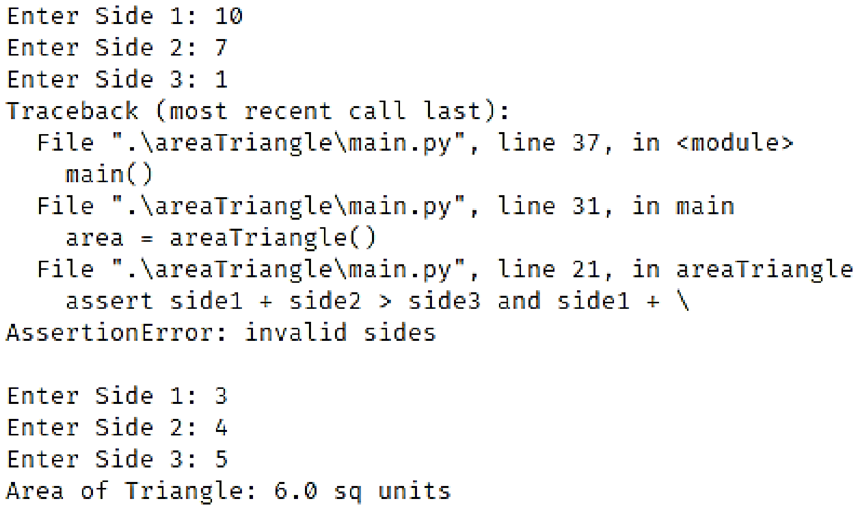
Objective

Write a function that takes the lengths of three sides: side1, side2 and side3 of the triangle as the input from the user using input function and return the area of the triangle as the output. Also, assert that sum of the length of any two sides is greater than the third side.

Code

```python
import math

def areaTriangle():
    '''
    Calculates the Area of a Triangle using Heron's Formula
    '''
    side1 = int(input('Enter Side 1: '))
    side2 = int(input('Enter Side 2: '))
    side3 = int(input('Enter Side 3: '))
    assert side1 + side2 > side3 and side1 + \
        side3 > side2 and side2 + side3 > side1, 'invalid sides'
    semiPerimeter = (side1 + side2 + side3) / 2
    return math.sqrt(semiPerimeter *
                     (semiPerimeter - side1) *
                     (semiPerimeter - side2) *
                     (semiPerimeter - side3))

def main():
    area = areaTriangle()
    if area:
        print('Area of Triangle:', area, 'sq units')

if_name_== '_main__':
    main()
```

Output



```
Enter Side 1: 10
Enter Side 2: 7
Enter Side 3: 1
Traceback (most recent call last):
  File ".\areaTriangle\main.py", line 37, in <module>
    main()
  File ".\areaTriangle\main.py", line 31, in main
    area = areaTriangle()
  File ".\areaTriangle\main.py", line 21, in areaTriangle
    assert side1 + side2 > side3 and side1 + \
AssertionError: invalid sides

Enter Side 1: 3
Enter Side 2: 4
Enter Side 3: 5
Area of Triangle: 6.0 sq units
```

PRACTICAL 2

Objective

Consider a showroom of electronic products, where there are various salesmen. Each salesman is given a commission of 5%, depending on the sales made per month. In case the sale done is less than 50000, then the salesman is not given any commission. Write a function to calculate total sales of a salesman in a month, commission and remarks for the salesman. Sales done by each salesman per week is to be provided as input. Use tuples/list to store data of salesmen.

Assign remarks according to the following criteria:

- Excellent: Sales >=80000
- Good: Sales>=60000 and <80000
- Average: Sales>=40000 and <60000
- Work Hard: Sales < 40000

Code

```
def calculateRenumeration(n):
    '''
    Calculates sales, commission and determines
    the remarks for n salesmen

    Accepts:
        n {int} -- number of salesmen
    '''
    s = 0
    salesmen = []
    for i in range(1, n + 1, 1):
        salesman = [0, 0, '']
        print(f'\nSalesman {i}')
        print('============')
        for j in range(1, 5, 1):
            s = float(input(f'Enter Sales in Week {j}: '))
            assert s >= 0, 'invalid entry'
            salesman[0]  +=  s
        if salesman[0] > 50000:
            salesman[1] = 0.05 * salesman[0]
        if salesman[0] >= 80000:
            salesman[2] = 'Excellent'
        elif salesman[0] >= 60000:
            salesman[2] = 'Good'
        elif salesman[0] >= 40000:
            salesman[2] = 'Average'
        elif salesman[0] < 40000:
            salesman[2] = 'Work Hard'
        salesmen.append(salesman)
    print()
    for i in range(1, n + 1, 1):
        print('''
    Salesman %d Summary
================================
Total   Sales:    %10.2f
Total Commission: %10.2f
Remarks:  %10s
        ''' % (i,
               salesmen[i - 1][0],
               salesmen[i - 1][1],
               salesmen[i - 1][2]))

def main():
    n = 0
    n = int(input('Enter Number of Salesmen: '))
    calculateRenumeration(n)

if_name_== '_main__':
    main()
```

Output

```
Enter Number of Salesmen: 2

Salesman 1
============
Enter Sales in Week 1: 1000.32
Enter Sales in Week 2: 2132
Enter Sales in Week 3: 678.2
Enter Sales in Week 4: 9000

Salesman 2
============
Enter Sales in Week 1: 20000
Enter Sales in Week 2: 50500
Enter Sales in Week 3: 50905
Enter Sales in Week 4: 230

    Salesman 1 Summary
================================
Total Sales:        12810.52
Total Commission:       0.00
Remarks:           Work Hard

    Salesman 2 Summary
================================
Total Sales:       121635.00
Total Commission:    6081.75
Remarks:           Excellent
```

PRACTICAL 3

Objective

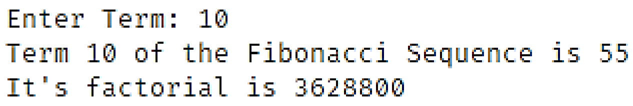
Write a Python function to find the nth term of Fibonacci sequence and its factorial. Return the result as a list.

Code

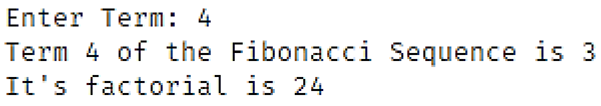
```python
def fibonacci(n):
    '''
    Calculates nth term of Fibonacci sequence

    Arguments:
        n {integer} -- term
    Returns:
        term {integer}
    '''
    assert n > 0, 'invalid index'
    if n == 1 or n == 2:
        return 1
    else:
        return fibonacci(n - 1) + fibonacci(n - 2)

def factorial(n):
    '''
    Calculates factorial of a number

    Accepts:
        n {int} -- input
    Returns:
        factorial {int}
    '''
    assert n >= 0, 'invalid number'
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)

def fibonacciAndFactorial(n):
    '''
    Returns the nth term of Fibonacci sequence
    and its factorial

    Accepts:
        n {int} -- an integer
    Returns
```

```
        [term, factorial] {list}
    '''
    return [fibonacci(n), factorial(n)]

def main():
    n = int(input('Enter Term: '))
    f = fibonacciAndFactorial(n)
    print(f'Term {n} of the Fibonacci Sequence is {f[0]}')
    print(f'It\'s factorial is {f[1]}')

if_name_== '_main__':
    main()
```

Output



```
Enter Term: 10
Term 10 of the Fibonacci Sequence is 55
It's factorial is 3628800
```



```
Enter Term: 4
Term 4 of the Fibonacci Sequence is 3
It's factorial is 24
```

PRACTICAL 4

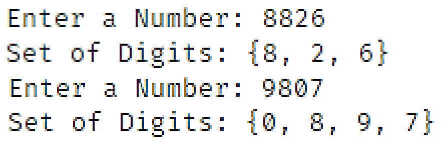
Objective

Write a function that takes a number (>=10) as an input and return the digits of the number as a set.

Code

```
def setOfDigits(n):
    '''
    Takes a number and returns its digits
    as a set

    Accepts:
      n {int} -- a number
    Returns:
      digits {set} -- digits of the number
    '''
    assert n >= 10
    s = set()
    while n != 0:
        s.add(n % 10)
        n //= 10
    return s

def main():
    n = int(input('Enter a Number: '))
    print('Set of Digits:', setOfDigits(n))

if_name_== '_main__':
    main()
```

Output



```
Enter a Number: 8826
Set of Digits: {8, 2, 6}
Enter a Number: 9807
Set of Digits: {0, 8, 9, 7}
```

PRACTICAL 5

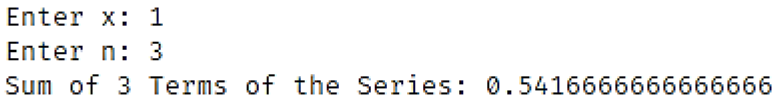
Objective

Write a function that finds the sum of the n terms of the series. Import the factorial function created in Q4.

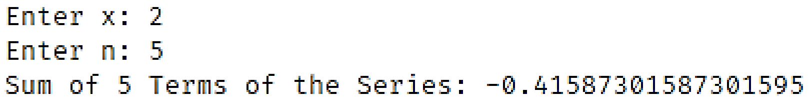
Code

```
# main.py
from factorial import factorial

def seriesSum(x, n):
    '''
    Calculates sum of n terms of the series

    Arguments:
        x {numeric} -- value of x
        n {integer} -- number of terms
    Returns:
        s {integer} -- sum
    '''
    assert n > 0, 'invalid number'
    e = 2
    s = 1
    sign = 1
    for i in range(n - 1):
        sign *= -1
        s += sign * (x ** e) / factorial(e)
        e += 2
    return s

def main():
    x = float(input('Enter x: '))
    n = int(input('Enter n: '))
    print(f'Sum of {n} Terms of the Series:', seriesSum(x, n))

if_name_== "_main__":
    main()

# factorial.py
def factorial(n):
    '''
    Calculates factorial of a number
    Accepts:
        n {integer} -- input
    Returns:
        factorial {integer}
    '''
    assert n >= 0, 'invalid number'
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)
```

Output



```
Enter x: 1
Enter n: 3
Sum of 3 Terms of the Series: 0.5416666666666666
```



```
Enter x: 2
Enter n: 5
Sum of 5 Terms of the Series: -0.41587301587301595
```

PRACTICAL 6

Objective

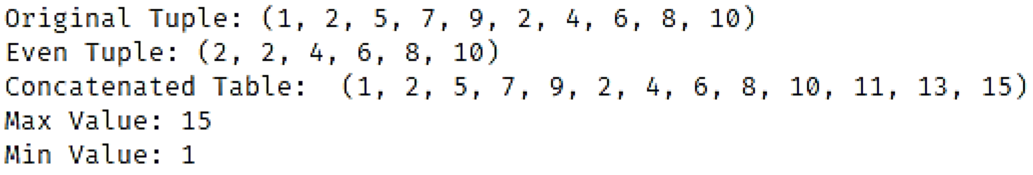
Consider a tuple t1 = {1,2,5,7,9,2,4,6,8,10}. Write a program to perform following operations:

(a) Print another tuple whose values are even numbers in the given tuple.
(b) Concatenate a tuple t2 = {11,13,15} with t1.
(c) Return maximum and minimum value from this tuple.

Code

```
def main():
    # Given
    t1 = (1, 2, 5, 7, 9, 2, 4, 6, 8, 10)
    t2 = (11, 13, 15)
    even_tuple = ()
    concatenated_tuple = ()
    max_value = 0
    min_value = 0
    print('Original Tuple:', t1)
    # Another Tuple with Even Numbers in t1
    for i in t1:
        if i % 2 == 0:
            even_tuple += (i,)
    print('Even Tuple:', even_tuple)
    # Concatenate t2
    concatenated_tuple = t1 + t2
    print('Concatenated Table: ', concatenated_tuple)
    # Max and Min Values in concatenated tuple
    print('Max Value:', max(concatenated_tuple))
    print('Min Value:', min(concatenated_tuple))

if_name_== '_main__':
    main()
```

Output



```
Original Tuple: (1, 2, 5, 7, 9, 2, 4, 6, 8, 10)
Even Tuple: (2, 2, 4, 6, 8, 10)
Concatenated Table:  (1, 2, 5, 7, 9, 2, 4, 6, 8, 10, 11, 13, 15)
Max Value: 15
Min Value: 1
```

PRACTICAL 7

Objective

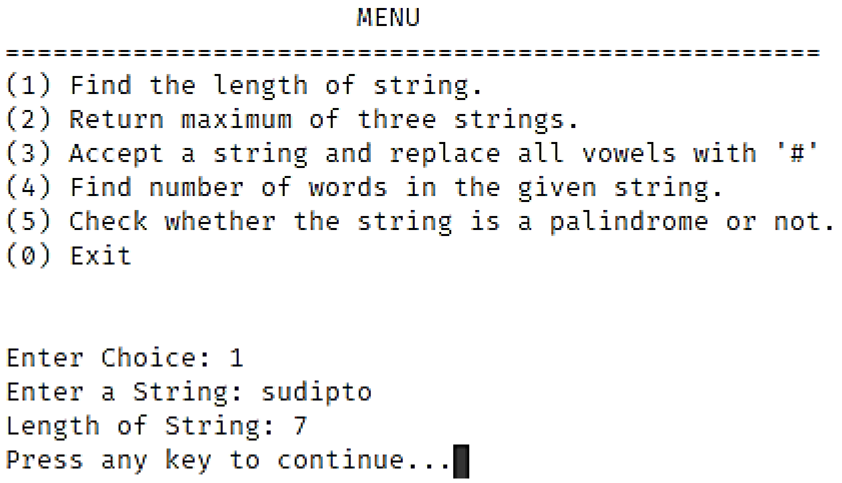
Write a menu driven program to perform the following on strings:

(a) Find the length of string.
(b) Return maximum of three strings.
(c) Accept a string and replace all vowels with '#'.
(d) Find number of words in the given string.
(e) Check whether the string is a palindrome or not.

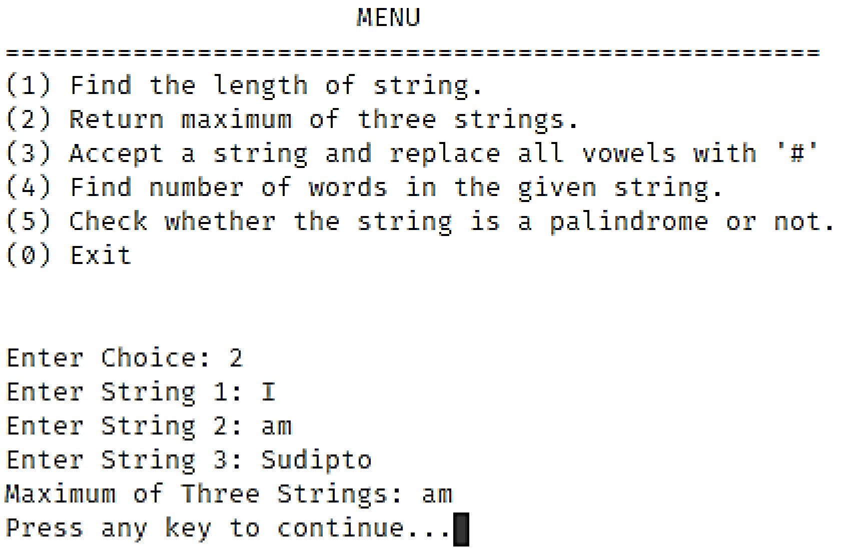
Code

```
def findLength():
    c = 0
    s = input('Enter a String: ')
    for i in s:
        c += 1
    print('Length of String:', c)

def maxOfStrings():
    maximum = ''
    s1 = input('Enter String 1: ')
    s2 = input('Enter String 2: ')
    s3 = input('Enter String 3: ')
    if s1 >= s2 and s1 >= s3:
        maximum = s1
    elif s2 >= s1 and s2 >= s3:
        maximum = s2
    else:
        maximum = s3
    print('Maximum of Three Strings:', maximum)

def replaceVowels():
    s = input('Enter a String: ')
    for i in s:
        if i in 'aAeEiIoOuU':
            s = s.replace(i, '#')
    print('Modified String:', s)

def numberOfWords():
    s = input('Enter a String: ')
    c = 0
    for i in s:
        if i == ' ':
            c += 1
    print('Number of Words:',  c + 1)

def isPalindrome():
    f = True
    s = input('Enter a String: ')
    for i in range(0, len(s) // 2, 1):
        if s[i] != s[len(s) - i - 1]:
            f = False
            print('String is not a Palindrome')
            break
    if f == True:
        print('String is a Palindrome')

def main():
    s = ''
    s1 = ''
    s2 = ''
    s3 = ''
    flag = 0
    while True:
        print('''
                      MENU
===================================================
(1) Find the length of string.
(2) Return maximum of three strings.
(3) Accept a string and replace all vowels with '#'
(4) Find number of words in the given string.
(5) Check whether the string is a palindrome or not.
(0) Exit

        ''')
        c = int(input('Enter Choice: '))
        if c == 1:
            findLength()
        elif c == 2:
            maxOfStrings()
        elif c == 3:
            replaceVowels()
        elif c == 4:
            numberOfWords()
        elif c == 5:
            isPalindrome()
        elif c == 0:
            break
        input('Press any key to continue...')

if_name_== '_main__':
    main()
```

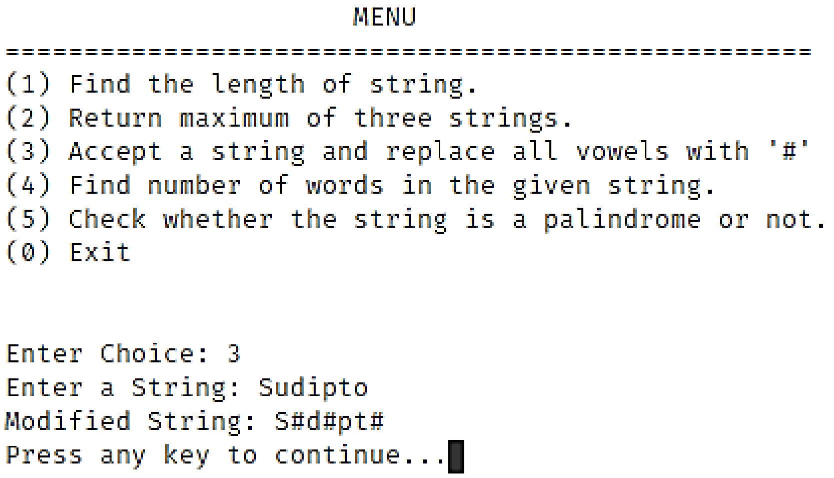
Output



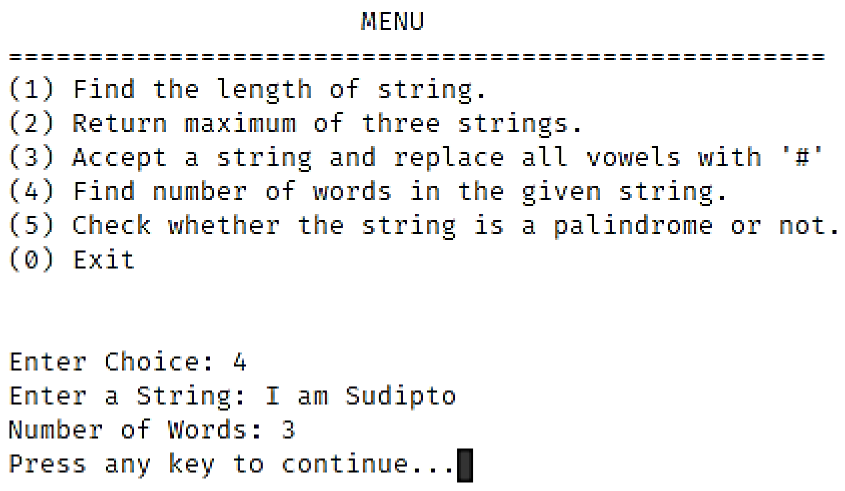
```
                      MENU
===================================================
(1) Find the length of string.
(2) Return maximum of three strings.
(3) Accept a string and replace all vowels with '#'
(4) Find number of words in the given string.
(5) Check whether the string is a palindrome or not.
(0) Exit


Enter Choice: 1
Enter a String: sudipto
Length of String: 7
Press any key to continue...
```



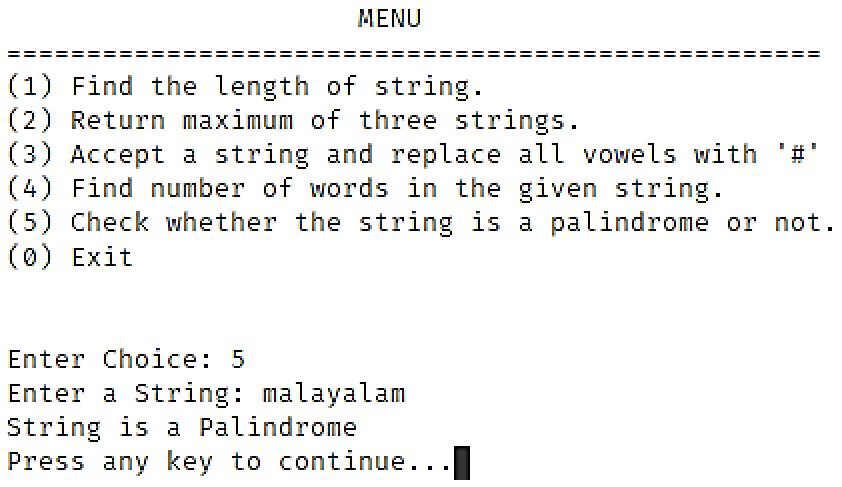
```
                      MENU
===================================================
(1) Find the length of string.
(2) Return maximum of three strings.
(3) Accept a string and replace all vowels with '#'
(4) Find number of words in the given string.
(5) Check whether the string is a palindrome or not.
(0) Exit


Enter Choice: 2
Enter String 1: I
Enter String 2: am
Enter String 3: Sudipto
Maximum of Three Strings: am
Press any key to continue...
```



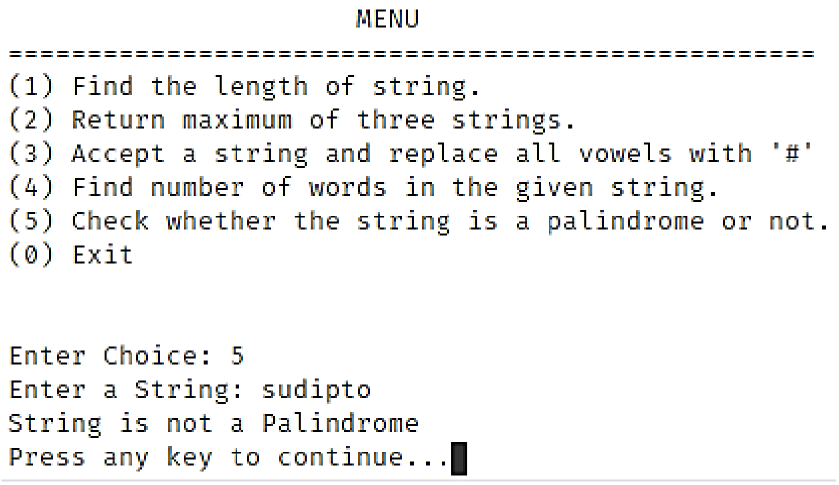
```
                      MENU
===================================================
(1) Find the length of string.
(2) Return maximum of three strings.
(3) Accept a string and replace all vowels with '#'
(4) Find number of words in the given string.
(5) Check whether the string is a palindrome or not.
(0) Exit


Enter Choice: 3
Enter a String: Sudipto
Modified String: S#d#pt#
Press any key to continue...
```



```
                        MENU
====================================================
(1) Find the length of string.
(2) Return maximum of three strings.
(3) Accept a string and replace all vowels with '#'
(4) Find number of words in the given string.
(5) Check whether the string is a palindrome or not.
(0) Exit


Enter Choice: 4
Enter a String: I am Sudipto
Number of Words: 3
Press any key to continue...
```



```
                        MENU
====================================================
(1) Find the length of string.
(2) Return maximum of three strings.
(3) Accept a string and replace all vowels with '#'
(4) Find number of words in the given string.
(5) Check whether the string is a palindrome or not.
(0) Exit


Enter Choice: 5
Enter a String: malayalam
String is a Palindrome
Press any key to continue...
```



```
                        MENU
====================================================
(1) Find the length of string.
(2) Return maximum of three strings.
(3) Accept a string and replace all vowels with '#'
(4) Find number of words in the given string.
(5) Check whether the string is a palindrome or not.
(0) Exit


Enter Choice: 5
Enter a String: sudipto
String is not a Palindrome
Press any key to continue...
```

PRACTICAL 8

Objective

Write a Python program to perform the following using lists:

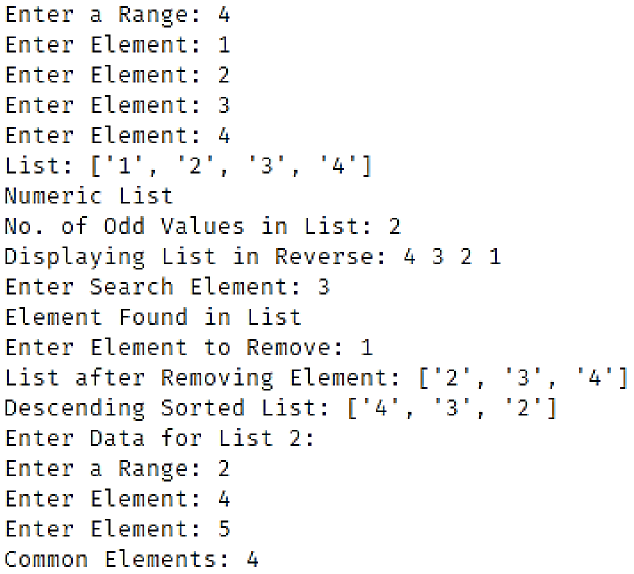
(a) Check if all elements in list are numbers or not.
(b) If it is a numeric list, then count number of odd values in it.
(c) If list contains all strings, then display largest string in the list.
(d) Display list in reverse form.
(e) Find a specified element in list.
(f) Remove the specified element from the list.
(g) Sort the list in descending order.
(h) Accept 2 lists and find the common members in them.

Code

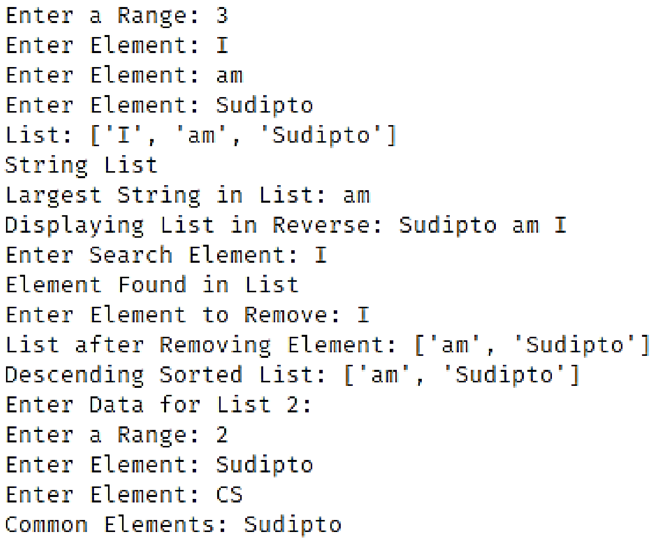
```
'''
Written by Sudipto Ghosh for the University of Delhi
'''
def isNumericList(l):
    for i in range(0, len(l), 1):
        if not l[i].isnumeric():
            return False
    return True

def isStringList(l):
    for i in range(0, len(l), 1):
        if not type(l[i]) == str:
            return False
    return True

def revDisplay(l):
    for i in range(len(l) - 1, -1, -1):
        print(l[i], end=' ')
    print()

def linearSearch(e, l):
    for i in range(0, len(l), 1):
        if l[i] == e:
            return True
    return False

def removeElement(e, l):
    if (linearSearch(e, l)):
        l.remove(e)
        return True
    return False
```

```
def revSort(l):
    for i in range(0, len(l), 1):
        for j in range(i + 1, len(l), 1):
            if l[i] < l[j]:
                l[i], l[j] = l[j], l[i]
    return l

def displayCommon(l1, l2):
    for i in l1:
        if i in l2:
            print(i, end=' ')
    print()

def main():
    l = []
    l2 = []
    r = 0
    e = 0
    c = 0
    r = int(input('Enter a Range: '))
    for i in range(0, r, 1):
        e = input('Enter Element: ')
        l.append(e)
    print('List:', end=' ')
    print(l)
    if (isNumericList(l)):
        print('Numeric List')
        for i in l:
            if not int(i) % 2 == 0:
                c += 1
        print('No. of Odd Values in List:', c)
    elif (isStringList(l)):
        print('String List')
        print('Largest String in List:', max(l))
    print('Displaying List in Reverse:', end=' ')
    revDisplay(l)
    e = input('Enter Search Element: ')
    if (linearSearch(e, l)):
        print('Element Found in List')
    else:
        print('Element Not Found in List')
    e = input('Enter Element to Remove: ')
    r = removeElement(e, l)
    if (r):
        print('List after Removing Element:', l)
    else:
        print('Element Not Found in List')
    print('Descending Sorted List:', revSort(l))

    print('Enter Data for List 2:')
    r = int(input('Enter a Range: '))
    for i in range(0, r, 1):
        e = input('Enter Element: ')
        l2.append(e)
    print('Common Elements:', end=' ')
    displayCommon(l, l2)

if_name_== '_main__':
    main()
```

Output



```
Enter a Range: 4
Enter Element: 1
Enter Element: 2
Enter Element: 3
Enter Element: 4
List: ['1', '2', '3', '4']
Numeric List
No. of Odd Values in List: 2
Displaying List in Reverse: 4 3 2 1
Enter Search Element: 3
Element Found in List
Enter Element to Remove: 1
List after Removing Element: ['2', '3', '4']
Descending Sorted List: ['4', '3', '2']
Enter Data for List 2:
Enter a Range: 2
Enter Element: 4
Enter Element: 5
Common Elements: 4
```



```
Enter a Range: 3
Enter Element: I
Enter Element: am
Enter Element: Sudipto
List: ['I', 'am', 'Sudipto']
String List
Largest String in List: am
Displaying List in Reverse: Sudipto am I
Enter Search Element: I
Element Found in List
Enter Element to Remove: I
List after Removing Element: ['am', 'Sudipto']
Descending Sorted List: ['am', 'Sudipto']
Enter Data for List 2:
Enter a Range: 2
Enter Element: Sudipto
Enter Element: CS
Common Elements: Sudipto
```