Solution,31 Jan,2022

1.Write a Python program to compute the product of the odd digits in a given number, or 0 if there aren't any.

Solution:

```
>>> def test(n):
>>>     if any(int(c) % 2 for c in str(n)):
>>>         prod = 1
>>>         for c in str(n):
>>>             if int(c) % 2 == 1:
>>>                 prod *= int(c)
>>>         return prod
>>>     return 0
>>>
>>>
>>> n = 123456789
>>> print("Original Number:",n)
>>> print("Product of the odd digits in the said number, or 0 if there aren't any")
>>> print(test(n))
>>> n = 2468
>>> print("\nOriginal Number:",n)
>>> print("Product of the odd digits in the said number, or 0 if there aren't any")
>>> print(test(n))
```

2. Write a Python program to find all n-digit integers that start or end with 2?

Solution:

```
>>> def test(n):
>>>     ans = []
>>>     for i in range(10 ** (n - 1), 10 ** n):
>>>         assert len(str(i)) == n
>>>         if str(i).startswith("2") or str(i).endswith("2"):
>>>             ans.append(i)
>>>     return ans
>>> n = 1
>>> print("Number:",n)
>>> print("All",n,"- digit integers that start or end with 2:")
```

```
>>> print(test(n))
>>> n = 2
>>> print("\nNumber:",n)
>>> print("All",n,"- digit integers that start or end with 2:")
>>> print(test(n))
>>> n = 3
>>> print("\nNumber:",n)
>>> print("All",n,"- digit integers that start or end with 2:")
>>> print(test(n))
```

3. Write a Python program to execute a string containing Python code. Go to the editor?

Solution:

```
>>> mycode = 'print("hello world")'
>>> code = """
>>> def mutiply(x,y):
>>>    return x*y

>>> print('Multiply of 2 and 3 is: ',mutiply(2,3))
"""
>>> exec(mycode)
>>> exec(code)
```

4.Write a Python program to find four positive even integers whose sum is a given integer

Solution:

```
>>> def test(n):
>>>     for a in range(n, 0, -1):
>>>         if not a % 2 == 0:
>>>             continue
>>>         for b in range(n - a, 0, -1):
>>>             if not b % 2 == 0:
>>>                 continue
>>>             for c in range(n - b - a, 0, -1):
>>>                 if not c % 2 == 0:
>>>                     continue
>>>                 for d in range(n - b - c - a, 0, >>> -1):
>>>                     if not d % 2 == 0:
```

```
>>>                     continue
>>>                 if a + b + c + d == n:
>>>                     return [a, b, c, d]
>>>
>>> n = 100
>>> print("Four positive even integers whose sum is",n)
>>> print(test(n))
>>> n = 1000
>>> print("\nFour positive even integers whose sum is",n)
>>> print(test(n))
>>> n = 10000
>>> print("\nFour positive even integers whose sum is",n)
>>> print(test(n))
>>> n = 1234567890
>>> print("\nFour positive even integers whose sum is",n)
>>> print(test(n))
```

5.write a Python program to flip a coin 1000 times and count heads and tails.

Solution:

```
>>> import random
>>> import itertools
>>>
>>> results = { 'heads': 0, 'tails': 0,}
>>> sides = list(results.keys())
>>>
>>> for i in range(10000):
>>>     results[random.choice(sides)] += 1

>>> print('Heads:', results['heads'])
>>> print('Tails:', results['tails'])
```

6.Write a Python program that accept name of given subject and marks. Input number of subjects in first line and subject name, marks separated by a space in next line. Print subject name and marks in order of its first occurrence.

Solution:

```
>>> import collections, re
>>> n = int(input("Number of subjects: "))
>>> item_order = collections.OrderedDict()
>>> for i in range(n):
```

```
>>>    sub_marks_list = re.split(r'(\d+)$',input("Input Subject name and marks: ").strip())
>>>    subject_name = sub_marks_list[0]
>>>    item_price = int(sub_marks_list[1])
>>>    if subject_name not in item_order:
>>> item_order[subject_name]=item_price
>>>    else:
>>>        >>>item_order[subject_name]=item_order[subject_name]+item_price

>>> for i in item_order:
>>>    print(i+str(item_order[i]))
```

7.write a binary search function which searches an item in a sorted list. The function should return the index of element to be searched in the list ?

Solution:

```
>>> import math
>>> def bin_search(li, element):
>>>     bottom = 0
>>>    top = len(li)-1
>>>    index = -1
>>>    while top>=bottom and index==-1:
>>>mid=int(math.floor((top+bottom)/2.0))
>>>        if li[mid]==element:
>>>            index = mid
>>>        elif li[mid]>element:
>>>            top = mid-1
>>>        else:
>>>            bottom = mid+1

>>>      return index

>>> li=[2,5,7,9,11,17,222]
>>> print (bin_search(li,11))
>>> print (bin_search(li,12))
```

8. Write a program to separate number and English alphabet character from string?

Solution:

```
>>> def splitString(str):

>>>    alpha = ""
>>>    num = ""
>>>    for i in range(len(str)):
>>>       if (str[i].isdigit()):
>>>          num = num+ str[i]
>>>       else:
>>>          alpha += str[i]

>>>    print(alpha)
>>>    print(num )


>>> if __name__ == "__main__":

>>>     str = "prerna12933dhingra56829"
>>>     splitString(str)
```

9. Write a Python function that accepts a string and calculate the number of upper case letters and lower case letters. Go to the editor
Sample String : 'The quick Brow Fox')

Solution:

```
>>> def string_test(s):
>>>     d={"UPPER_CASE":0, "LOWER_CASE":0}
>>>     for c in s:
>>>         if c.isupper():
>>>            d["UPPER_CASE"]+=1
>>>         elif c.islower():
>>>            d["LOWER_CASE"]+=1
>>>         else:
>>>            pass
>>>     print ("Original String : ", s)
>>>     print ("No. of Upper case characters : ", d["UPPER_CASE"])
>>>     print ("No. of Lower case Characters : ", d["LOWER_CASE"])

>>> string_test('The quick Brown Fox')
```

10.write a program to generate Fibonacci numbers series?

Solution:

```
>>> ##generating Fibonacci series
>>> n_terms = int(input ("How many terms the user wants to print? "))

>>> # First two terms
>>> n_1 = 0
>>> n_2 = 1
>>> count = 0

>>> # Now, we will check if the number of terms is valid or not
>>>if n_terms <= 0:
>>>     print ("Please enter a positive integer, the given number is not valid")
>>> # if there is only one term, it will >>  return n_1
>>> elif n_terms == 1:
>>>     print ("The Fibonacci sequence of the numbers up to", n_terms, ": ")
>>>     print(n_1)
>>> # Then we will generate Fibonacci sequence of number
>>> else:
>>>     print ("The fibonacci sequence of the numbers is:")
>>>     while count < n_terms:

>>>         print(n_1)
>>>         nth = n_1 + n_2
>>>        # At last, we will update values
>>>         n_1 = n_2

>>>        n_2 = nth
>>>      count += 1
```

.