1) The included code stub will read an integer, n, from STDIN.

Without using any string methods, try to print the following:123….N

Solution-if \_\_name\_\_ == '\_\_main\_\_':

    n = int(input())

for i in range(1,n+1):

    print(i,end="")

Graphical user interface, text, application

Description automatically generated

2) Both players are given the same string, .  
Both players have to make substrings using the letters of the string .  
Stuart has to make words starting with consonants.  
Kevin has to make words starting with vowels.  
The game ends when both players have made all possible substrings.

**Scoring**  
A player gets +1 point for each occurrence of the substring in the string .

Solution-

def minion\_game(string):

    vowel =['A','E','I','O','U']

    Stu=0

    Kev=0

    for i in range(len(string)):

        if string[i] in vowel:

            Kev+= len(string)-i

        else:

            Stu+=len(string)-i

    if Stu>Kev:

        print("Stuart"+" "+ "%d" % Stu)

    elif Kev>Stu:

        print("Kevin"+" "+'%d' % Kev)

    else:

        print("Draw")

if \_\_name\_\_ == '\_\_main\_\_':

    s = input()

    minion\_game(s)

Graphical user interface, application

Description automatically generated

3) Given an array of integers nums and an integer target, return indices of the two numbers such that they add up to *target*.

You may assume that each input would have **exactly one solution**, and you may not use the same element twice.

You can return the answer in any order.

Solution-

class Solution:

def twoSum(self, nums: List[int], target: int) -> List[int]:

for i in nums:

if target - i in nums[nums.index(i) + 1:]:

return [nums.index(i), nums[nums.index(i) + 1:].index(target - i) + nums.index(i) + 1]

A screenshot of a computer

Description automatically generated

4) Given the participants' score sheet for your University Sports Day, you are required to find the runner-up score. You are given  scores. Store them in a list and find the score of the runner-up.

Solution - if \_\_name\_\_ == '\_\_main\_\_':

    n = int(input())

    arr = map(int, input().split())

    print (sorted(set(arr))[-2])

Graphical user interface, text, application

Description automatically generated

5r oman numerals are represented by seven different symbols: I, V, X, L, C, D and M

Solution - class Solution:

def romanToInt(self, s: str) -> int:

dic = {'I':1,'V':5,'X':10,'L':50,'C':100,'D':500,'M':1000,'E':0}

if len(s) == 0:

return 0

if len(s) == 1:

return dic[s[0]]

if dic[s[0]] >= dic[s[1]]:

return dic[s[0]] + Solution.romanToInt(self, s[1:])

else:

diff = abs(dic[s[0]] - dic[s[1]])

return diff + Solution.romanToInt(self, s[2:])

Graphical user interface, text, application

Description automatically generated