

**UNIVERSITY INSTITUTE OF ENGINEERING**

**Department of Computer Science & Engineering**

**Subject Name:** Competitive Coding

**Subject Code:** 20CSP-314

**Submitted to: Submitted by:**

Er. Mamta Punia Name: Sahil Kaundal

UID: 21BCS8197

Section: 616

Group: A

**INDEX**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ex. No** | **List of Experiments** | **Conduct (MM: 12)** | **Viva**  **(MM: 10)** | **Record (MM: 8)** | **Total**  **(MM: 30)** | **Remarks/Signature** |
| 1 | To demonstrate the concept of Array. |  |  |  |  |  |
| 2 | To demonstrate the concept of Stack and Queue. |  |  |  |  |  |
| 3 | To demonstrate the concept of Linked List. |  |  |  |  |  |
| 4 | Sorting and Searching: Implement the concept of Searching and Sorting techniques. |  |  |  |  |  |
| 5 | To implement the concept of Graphs. |  |  |  |  |  |
| 6. | To demonstrate the concept of Tree Data Structure |  |  |  |  |  |
| 7. | To Demonstrate the concept of String Data Structure |  |  |  |  |  |
| 8. | Dynamic Programming |  |  |  |  |  |
|  |  |  |  |  |  |  |

**Experiment 8.1**

**Student Name:** Sahil Kaundal **UID:** 21BCS8197

**Branch:** BE CSE (Lateral Entry) **Section/Group:** 616/A

**Semester:** 5th **Date of Performance:** 04/11/2022

**Subject Name:** CC Lab **Subject Code:** 20CSP-314

1. **Aim/Overview of the practical:**

Dynamic Programming

Samantha and Sam are playing a numbers game. Given a number as a string, no leading zeros, determine the sum of all integer values of substrings of the string.

<https://www.hackerrank.com/challenges/sam-and-substrings/problem?isFullScreen=true>

1. **Apparatus / Simulator Used:**

* Windows 7 or above
* Google Chrome

1. **Objective:**
   * To understand the concept of Dynamic Programming.
   * To implement the concept of Dynamic Programming.
   * Goal is to find the number of ways to construct an array such that consecutive positions contain different values.

**4. Code:**

def solution(n):

    s = 0

    prev\_sum = 0

    for i, d in enumerate(n):

        s\_ = prev\_sum \* 10 + (i + 1) \* int(d)

        s += s\_

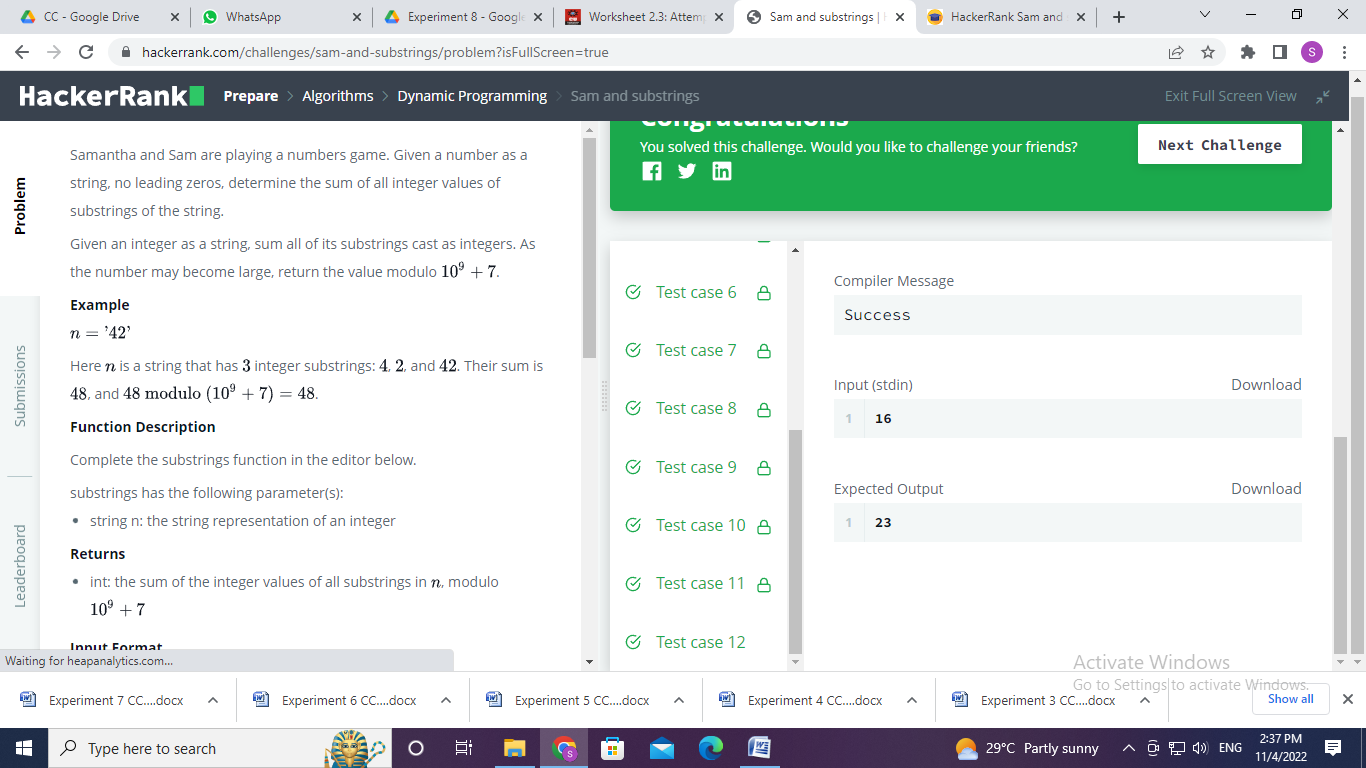
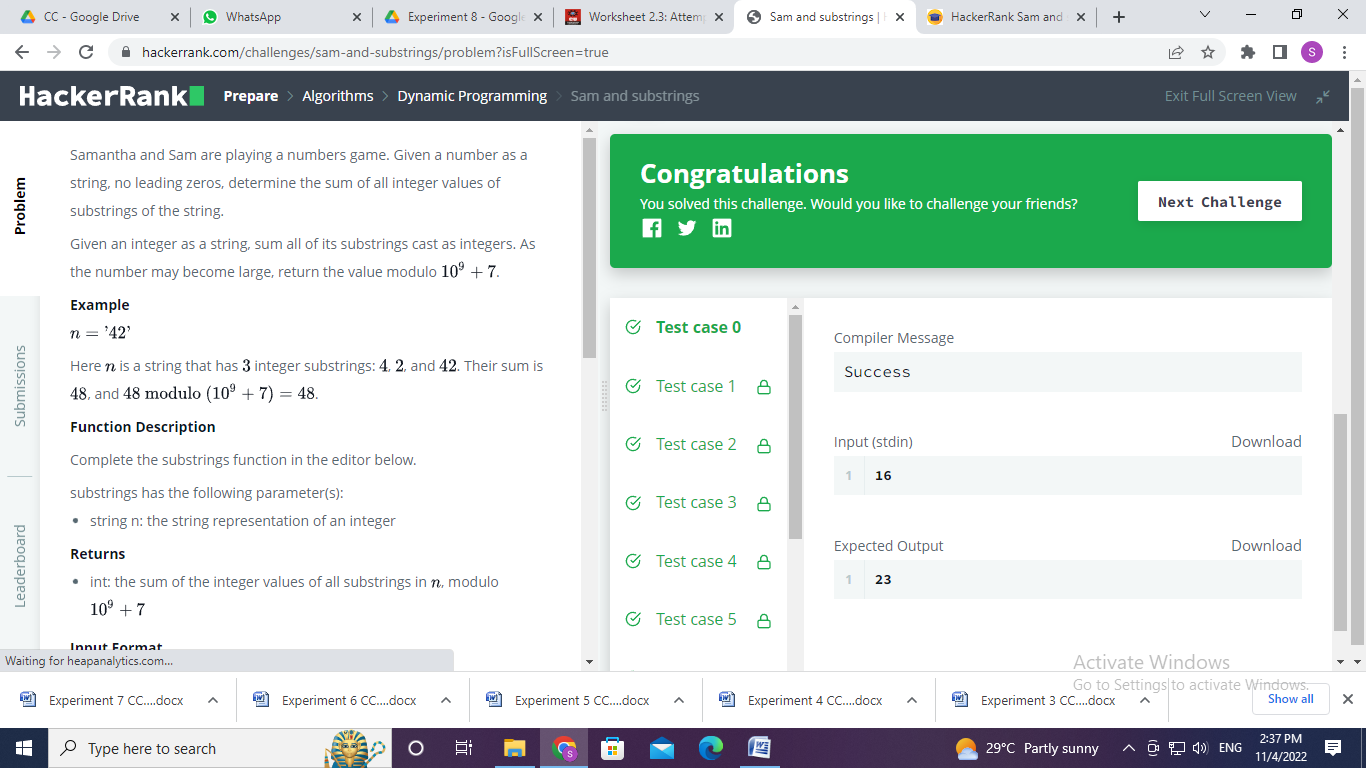
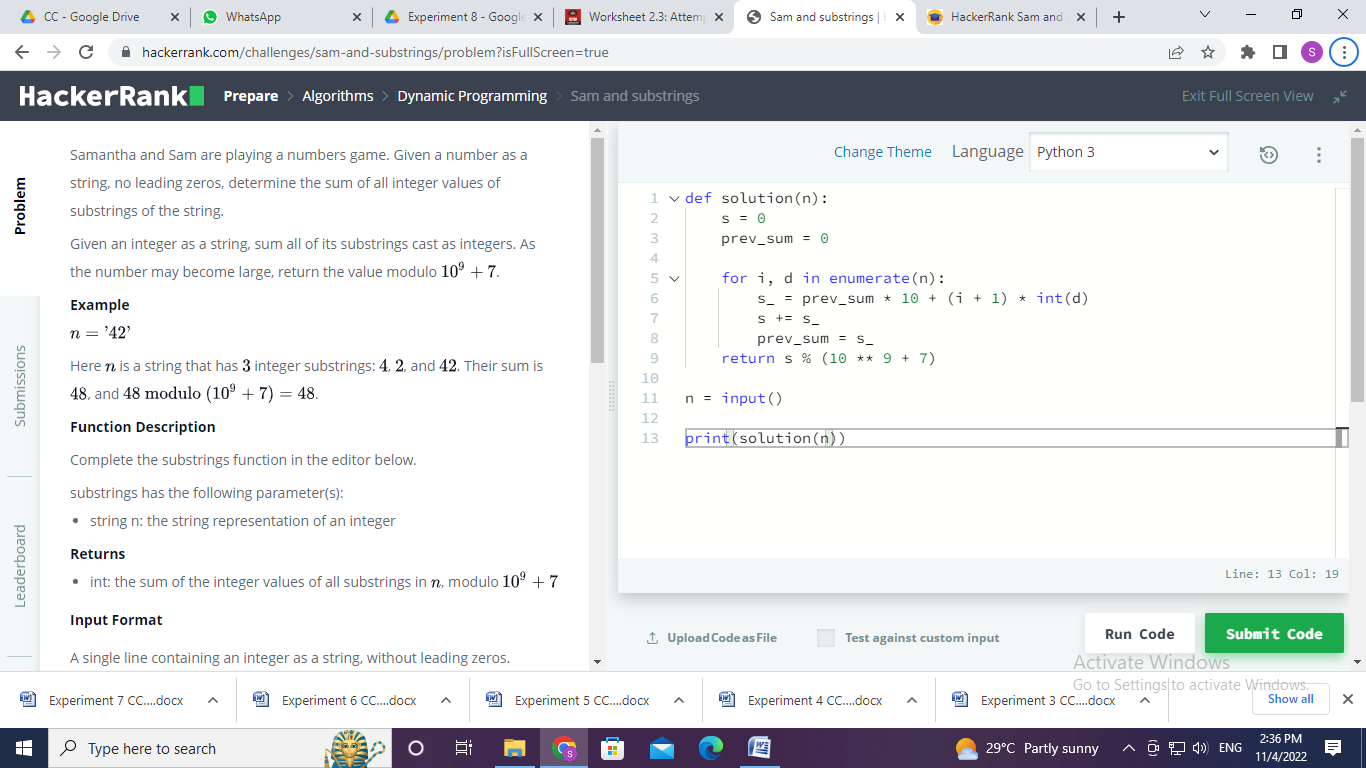
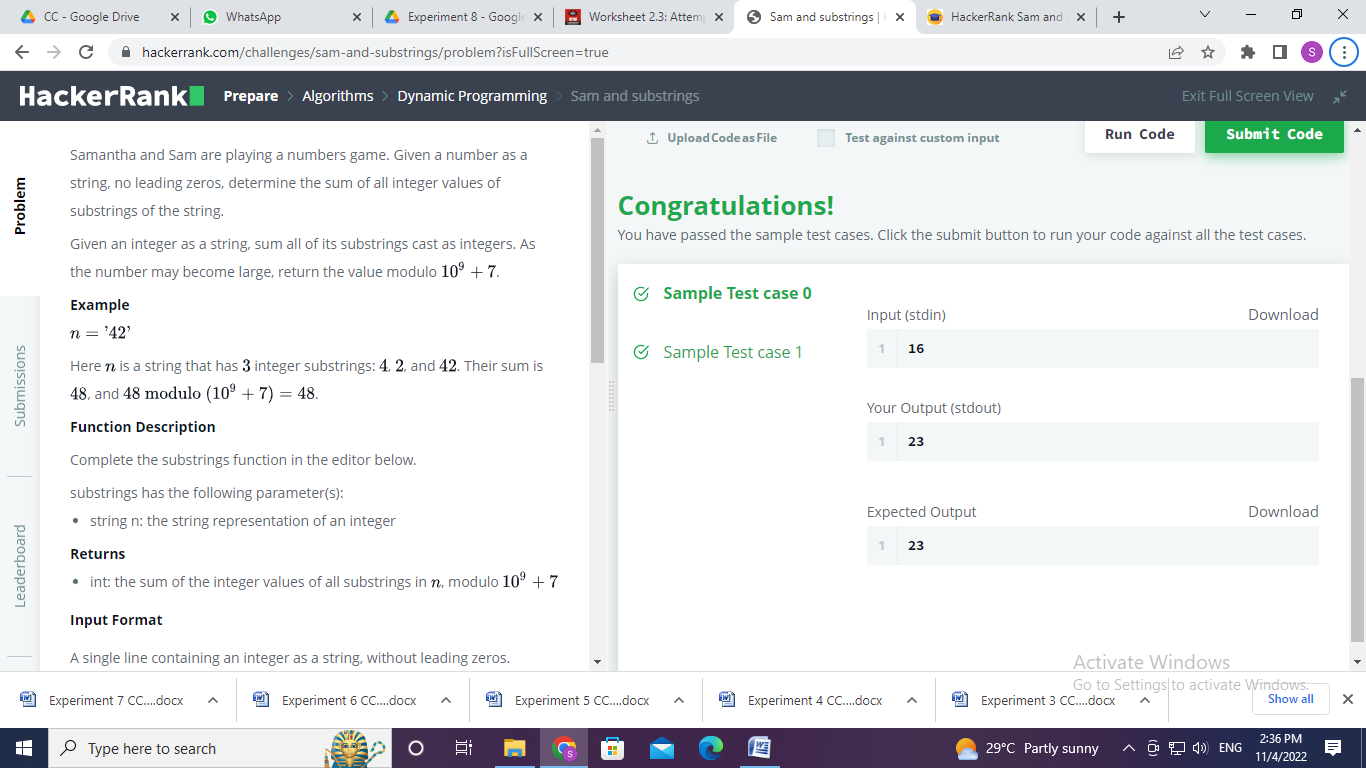
        prev\_sum = s\_

    return s % (10 \*\* 9 + 7)

n = input()

print(solution(n))

**5. Result/Output/Writing Summary:**



**Experiment 8.2**

1. **Aim/Overview of the practical:**

DynamicProgramming

Red John has committed another murder. This time, he doesn't leave a red smiley behind. Instead he leaves a puzzle for Patrick Jane to solve. He also texts Teresa Lisbon that if Patrick is successful, he will turn himself in.

<https://www.hackerrank.com/challenges/red-john-is-back/problem?isFullScreen=true>

1. **Apparatus / Simulator Used:**

* Windows 7 or above
* Google Chrome

1. **Objective:**
   * To understand the concept of Dynamic Programming.
   * To implement the concept of Dynamic Programming.
   * Goal is to find the number of ways to construct an array such that consecutive positions contain different values.
2. **Code:**

def primes(n):

    """ Returns  a list of primes < n """

    if n <= 2: return 0

    sieve = [True] \* n

    for i in range(3,int(n\*\*0.5)+1,2):

        if sieve[i]:

            sieve[i\*i::2\*i]=[False]\*int((n-i\*i-1)/(2\*i)+1)

    return len([i for i in range(3,n,2) if sieve[i]]) + 1

def find\_configs(N):

    if N == 0:

        return 1

    elif N < 0:

        return 0

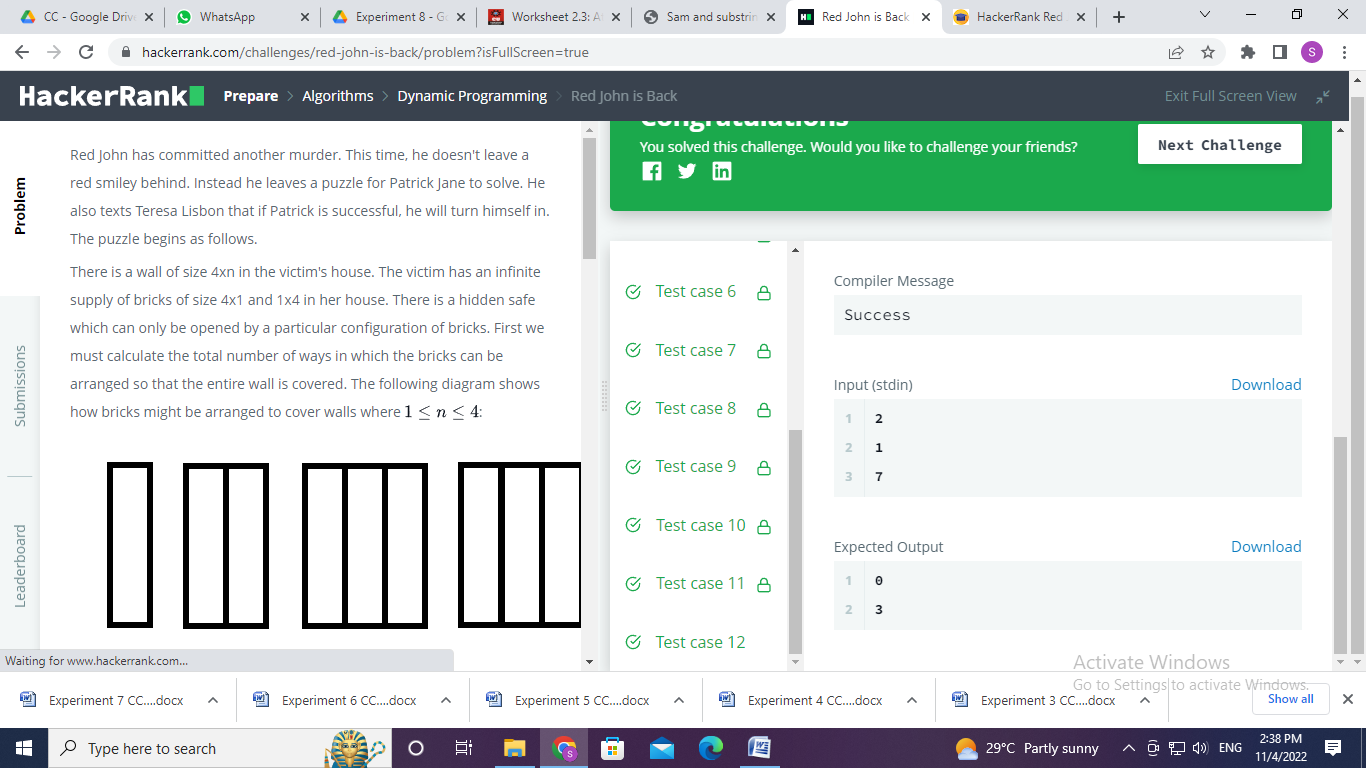
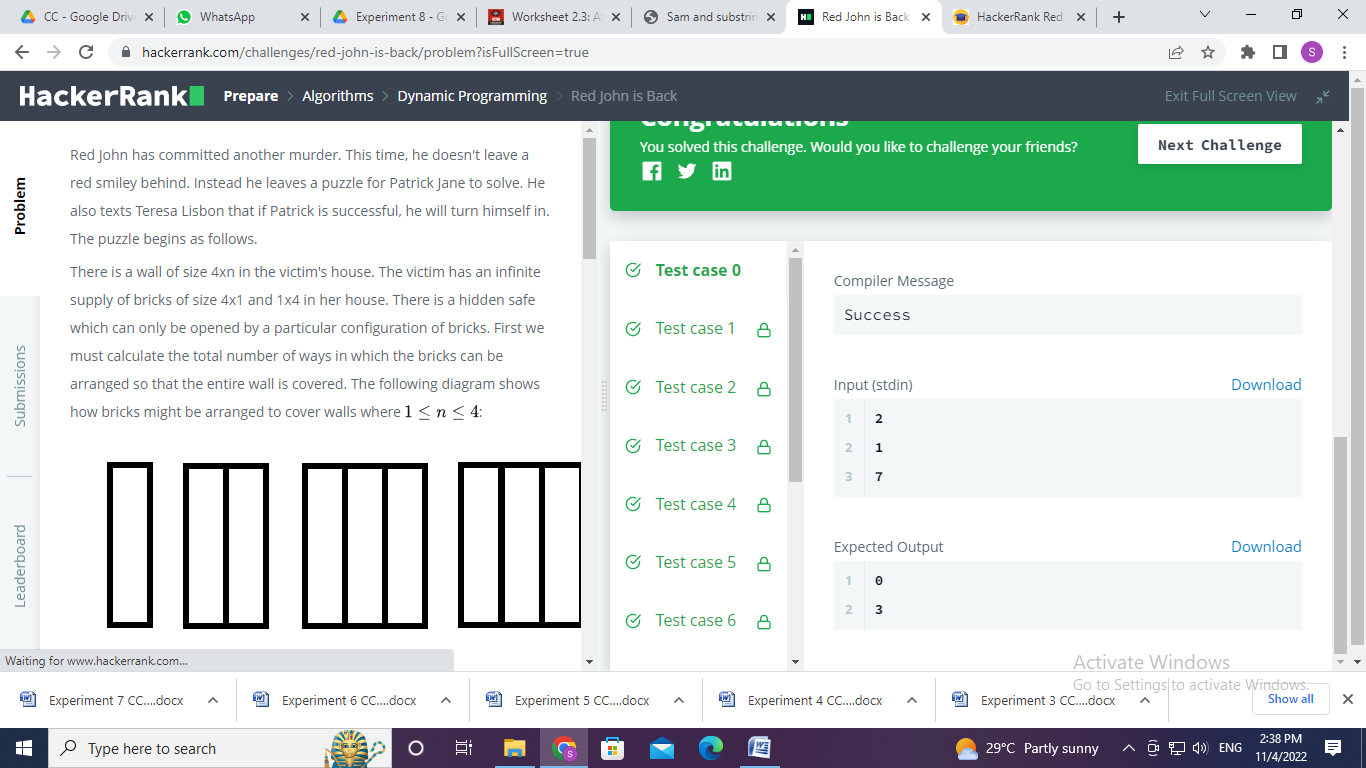
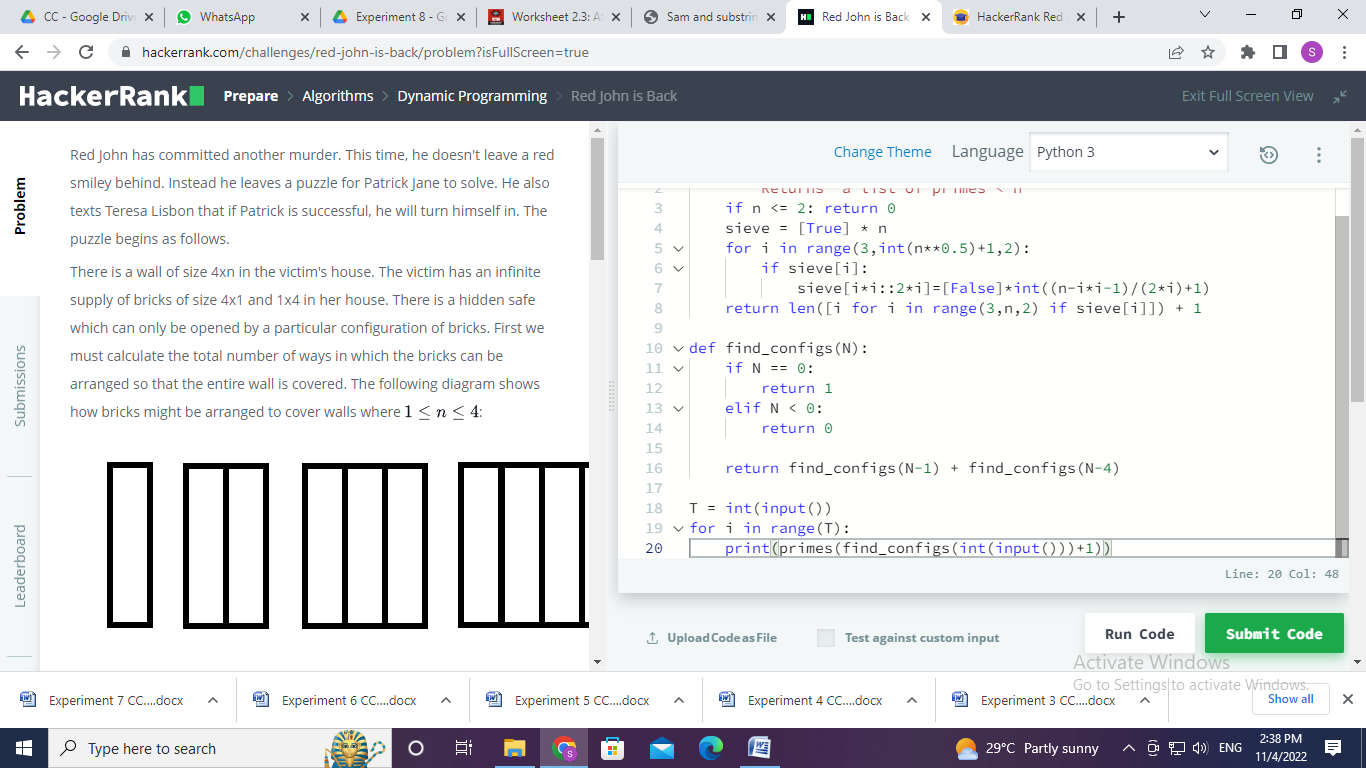
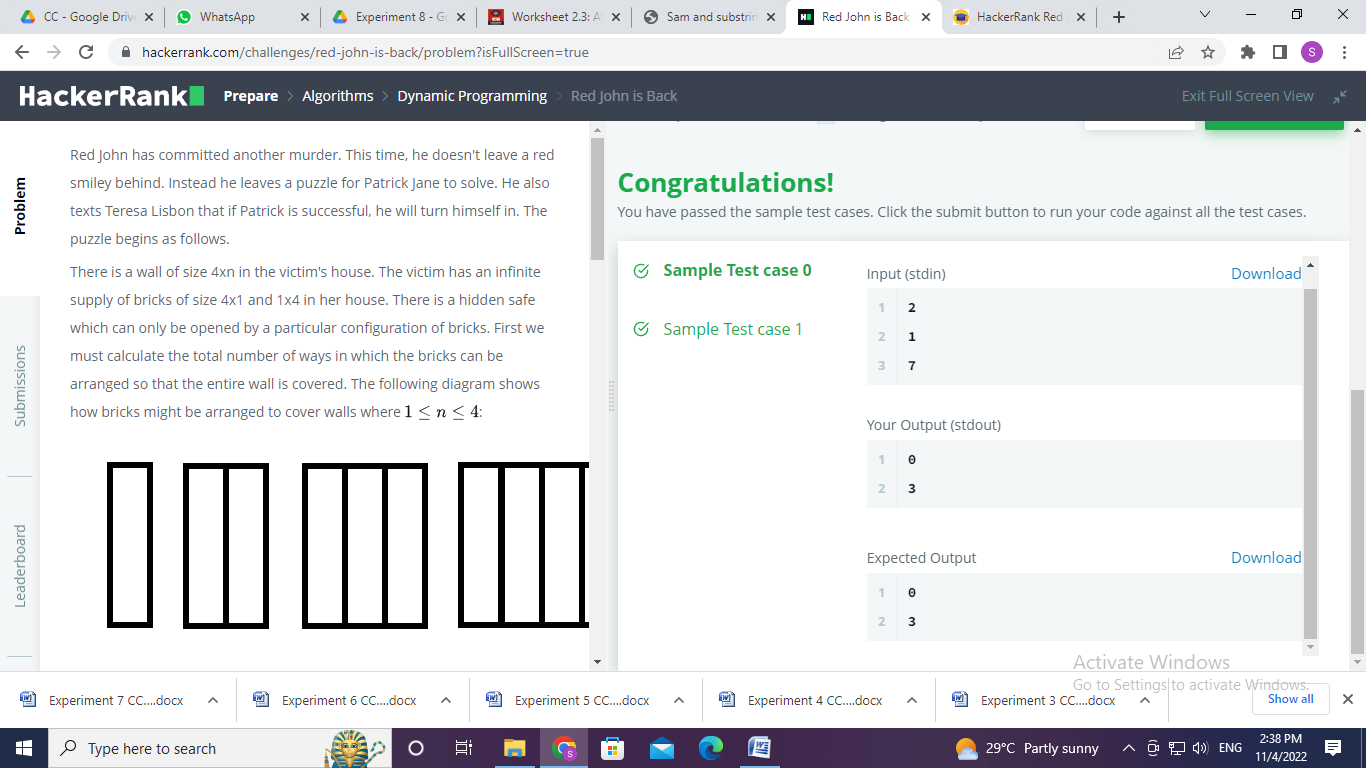
    return find\_configs(N-1) + find\_configs(N-4)

T = int(input())

for i in range(T):

    print(primes(find\_configs(int(input()))+1))

1. **Result/Output/Writing Summary:**



**Learning outcomes (What I have learnt):**

* + Learned the concept of Dynamic Programming.
  + Learnt about Array in Dynamic Programming.
  + Learn about the countArray and Equal concept.

**Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. |  |  |  |
| 2. |  |  |  |
| 3. |  |  |  |
|  |  |  |  |