CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

UNIVERSITY INSTITUTE OF ENGINEERING

Department of Computer Science & Engineering

Subject Name: Competitive Coding

Subject Code: 20CSP-314

Submitted by:

Name: Pranjal Deep

UID: 20BCS9286

Section: 619

Group: B

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

Student Name: Anjali Singh
UID: 20BCS9286
Branch: BE CSE
Section/Group: 619/B
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

2. **Apparatus / Simulator Used:**
   - Windows 7 or above
   - Google Chrome

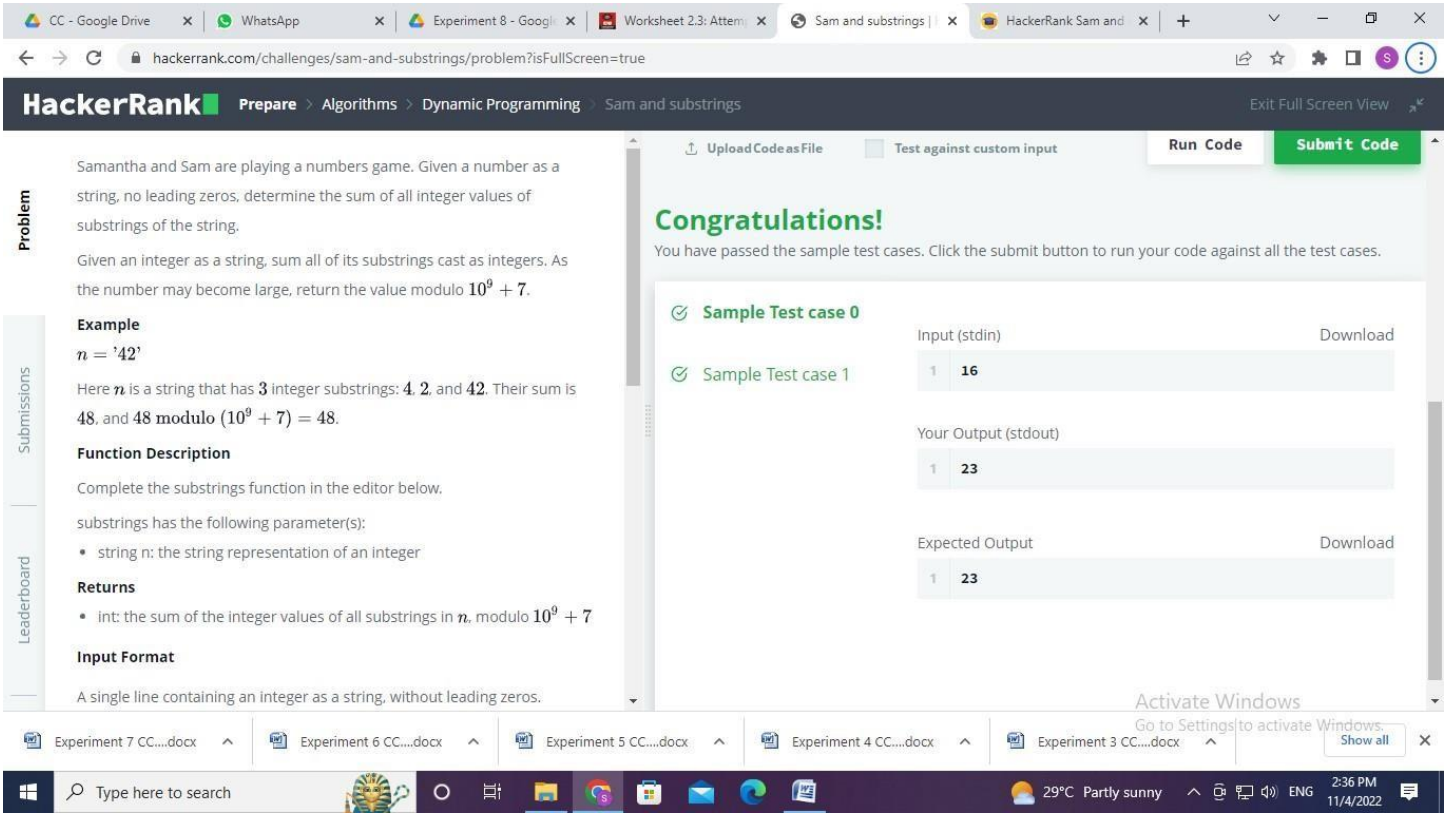
3. **Objective:**
   - To understand the concept of Dynamic Programming.
   - To implement the concept of Dynamic Programming.
   - Goal is to find the number of ways to construct an array such that consecutive positions contain different values.

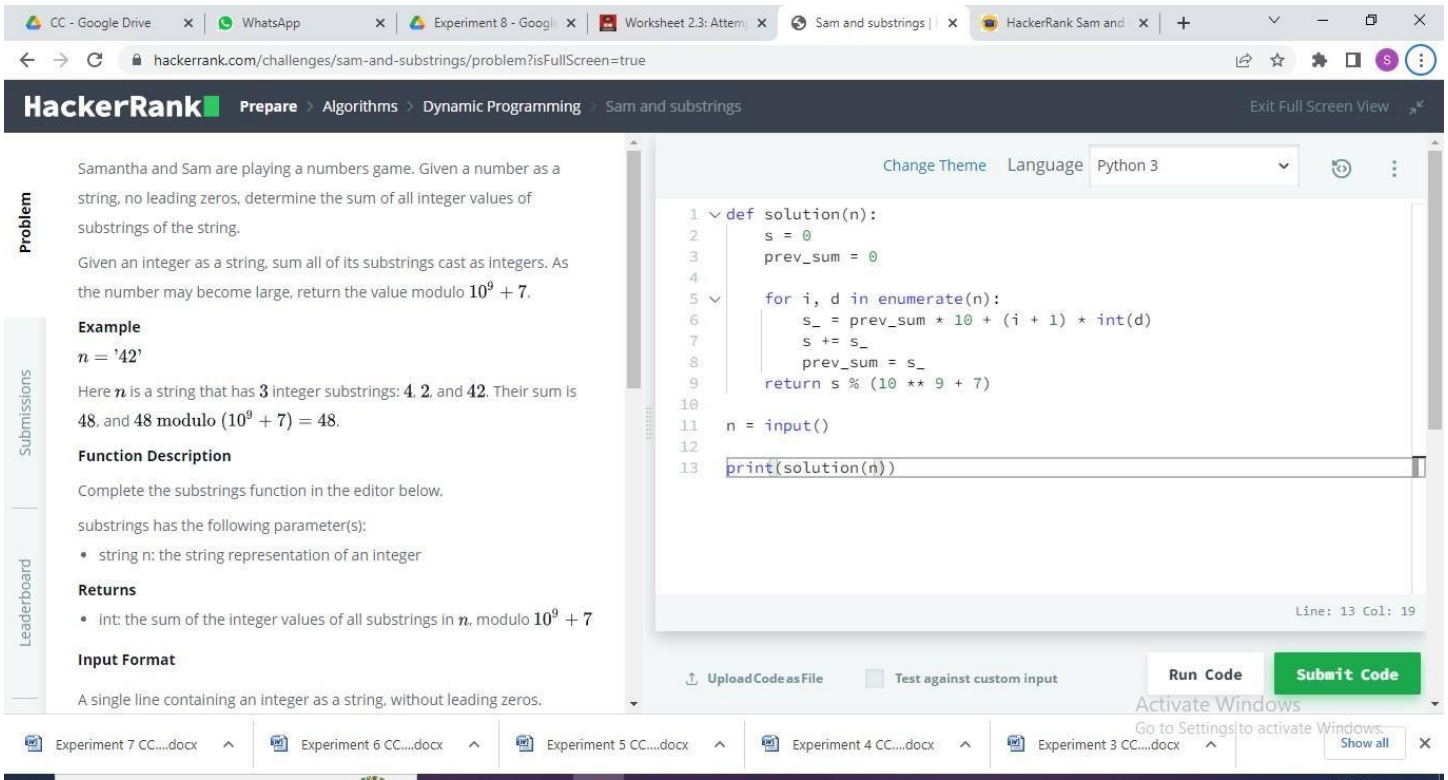
4. **Code:**

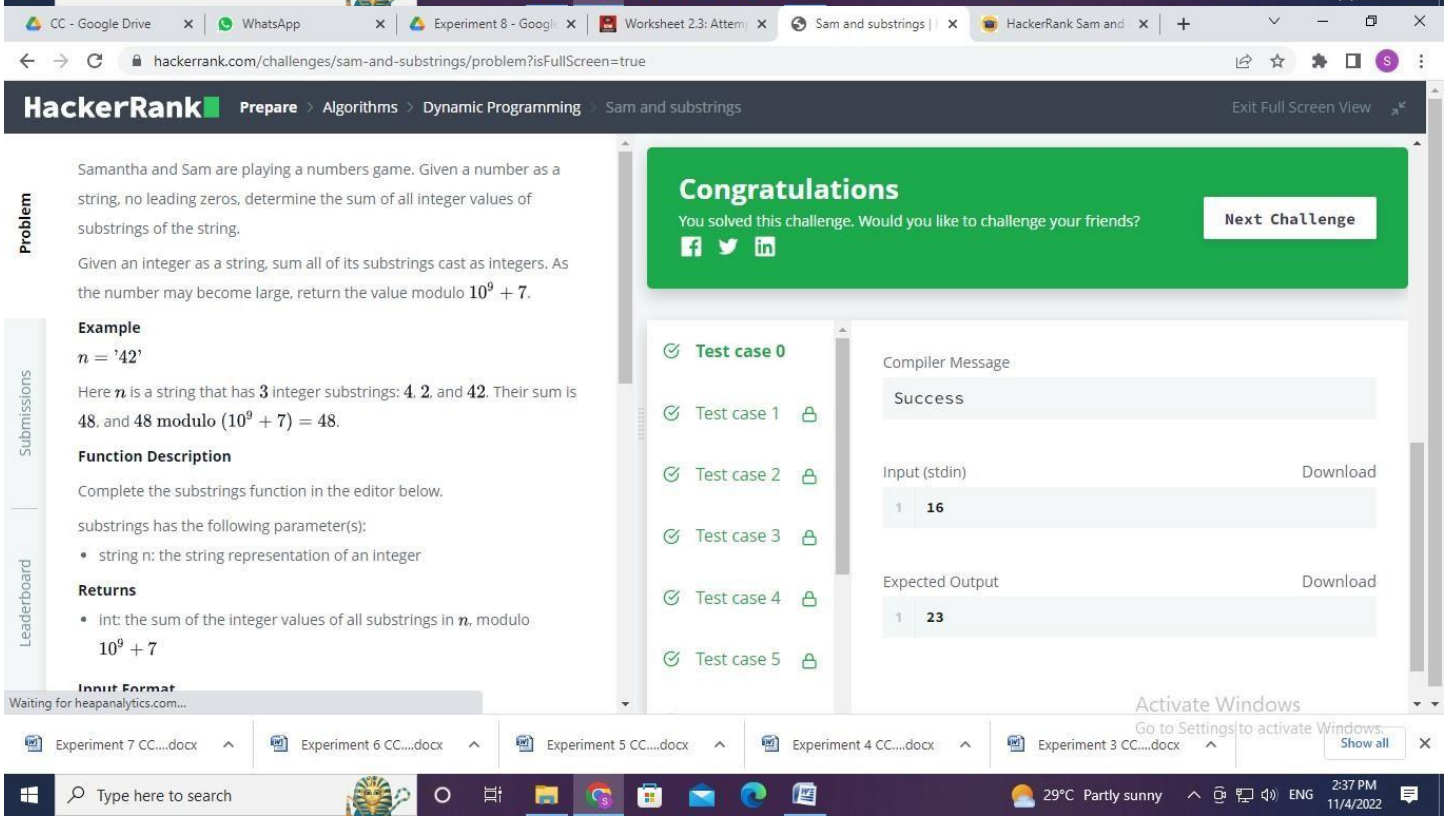
```
def solution(n):
s = 0     prev_sum
= 0
         for i, d in
enumerate(n):
        s_ = prev_sum * 10 + (i + 1) * int(d)
s += s_          prev_sum = s_     return s %
(10 ** 9 + 7)

n = input()
```

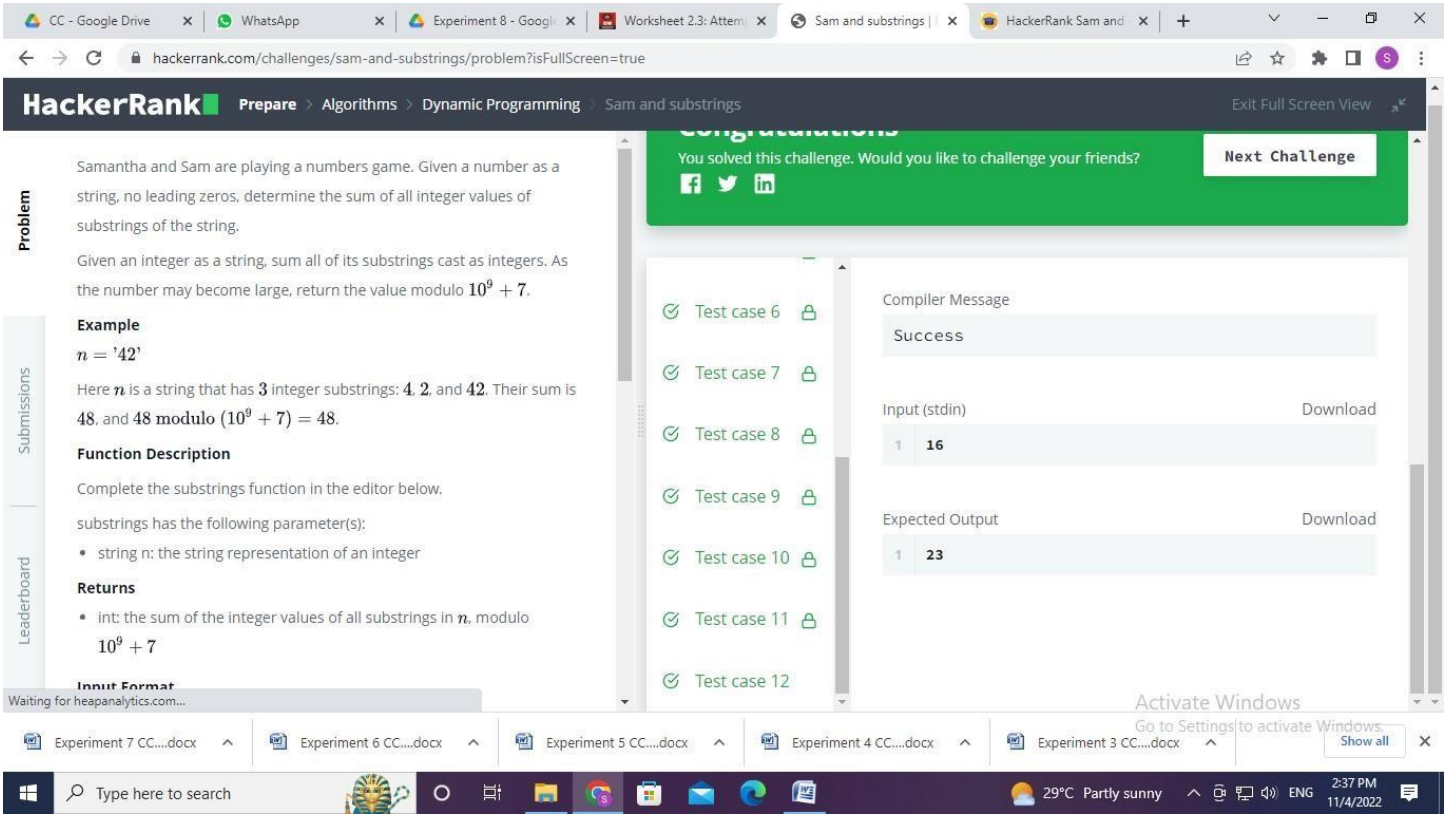
```
print(solution(n))
```

5. Result/Output/Writing Summary:




CC - Google Drive
WhatsApp
Experiment 8 - Google
Worksheet 2.3: Attem
Sam and substrings
HackerRank Sam and
hackerrank.com/challenges/sam-and-substrings/problem?isFullScreen=true
HackerRank
Prepare > Algorithms > Dynamic Programming > Sam and substrings
Exit Full Screen View
Problem
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.
Given an integer as a string, sum all of its substrings cast as integers. As the number may become large, return the value modulo $10^9 + 7$.
Example
$n = '42'$
Here n is a string that has 3 integer substrings: 4, 2, and 42. Their sum is 48, and 48 modulo $(10^9 + 7) = 48$.
Function Description
Complete the substrings function in the editor below.
substrings has the following parameter(s):
string n: the string representation of an integer
Returns
int: the sum of the integer values of all substrings in n, modulo $10^9 + 7$
Input Format
A single line containing an integer as a string, without leading zeros.
Submissions
Leaderboard
Change Theme
Language
Python 3
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))
Line: 13 Col: 19
Upload Code as File
Test against custom input
Run Code
Submit Code
Activate Windows
Go to Settings to activate Windows.
Experiment 7 CC....docx
Experiment 6 CC....docx
Experiment 5 CC....docx
Experiment 4 CC....docx
Experiment 3 CC....docx
Show all
Type here to search
29°C Partly sunny
ENG
2:36 PM
11/4/2022


CC - Google Drive
WhatsApp
Experiment 8 - Google
Worksheet 2.3: Attem
Sam and substrings
HackerRank Sam and
hackerrank.com/challenges/sam-and-substrings/problem?isFullScreen=true
HackerRank
Prepare > Algorithms > Dynamic Programming > Sam and substrings
Exit Full Screen View
Problem
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.
Given an integer as a string, sum all of its substrings cast as integers. As the number may become large, return the value modulo $10^9 + 7$.
Example
$n = '42'$
Here n is a string that has 3 integer substrings: 4, 2, and 42. Their sum is 48, and 48 modulo $(10^9 + 7) = 48$.
Function Description
Complete the substrings function in the editor below.
substrings has the following parameter(s):
string n: the string representation of an integer
Returns
int: the sum of the integer values of all substrings in n, modulo $10^9 + 7$
Input Format
Submissions
Leaderboard
Waiting for heapanalytics.com...
Congratulations
You solved this challenge. Would you like to challenge your friends?
Next Challenge
Test case 0
Test case 1
Test case 2
Test case 3
Test case 4
Test case 5
Compiler Message
Success
Input (stdin)
Download
1 16
Expected Output
Download
1 23
Activate Windows
Go to Settings to activate Windows.
Experiment 7 CC....docx
Experiment 6 CC....docx
Experiment 5 CC....docx
Experiment 4 CC....docx
Experiment 3 CC....docx
Show all
Type here to search
29°C Partly sunny
ENG
2:37 PM
11/4/2022



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

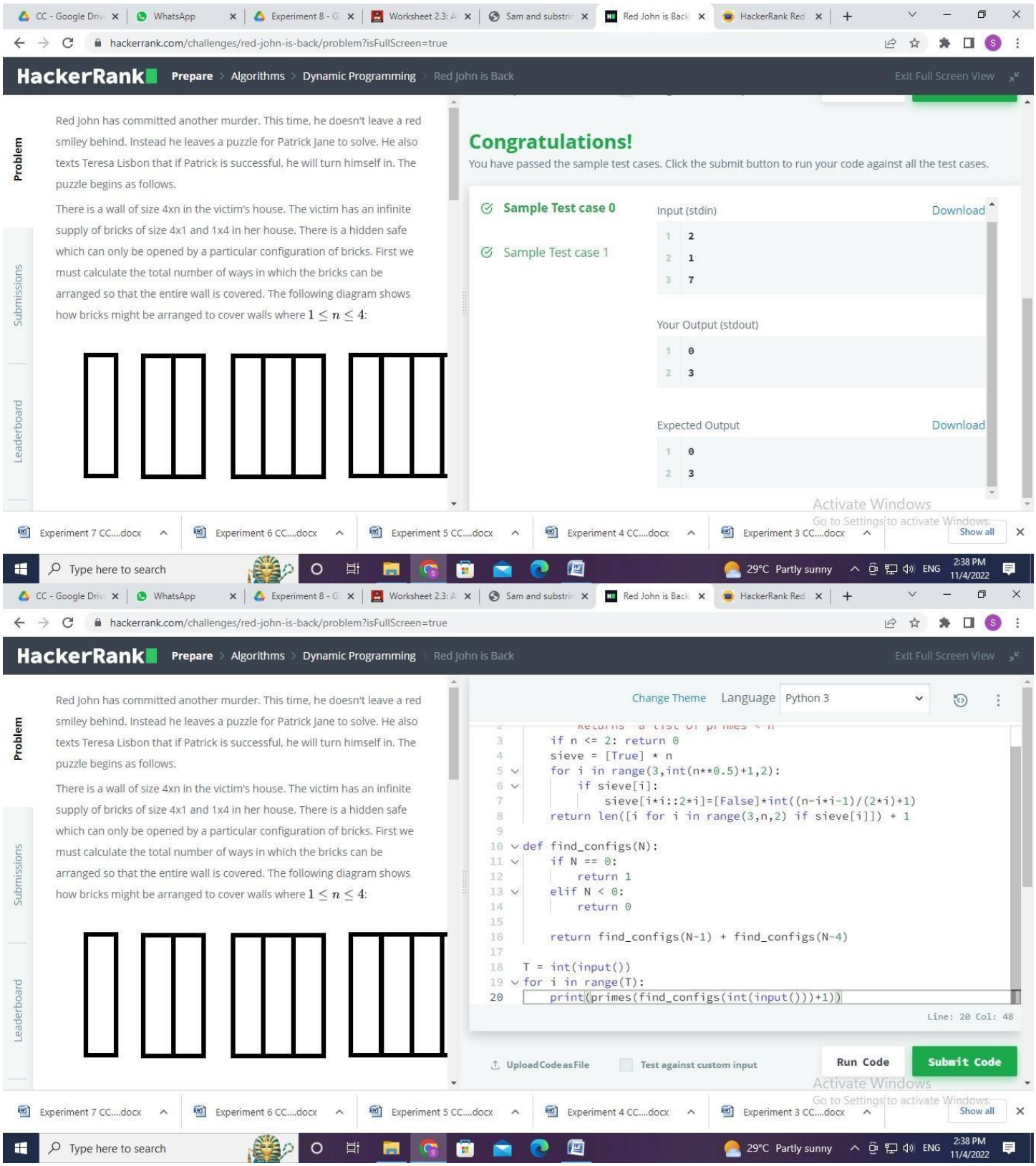
2. **Apparatus / Simulator Used:**
   - Windows 7 or above
   - Google Chrome

3. **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.

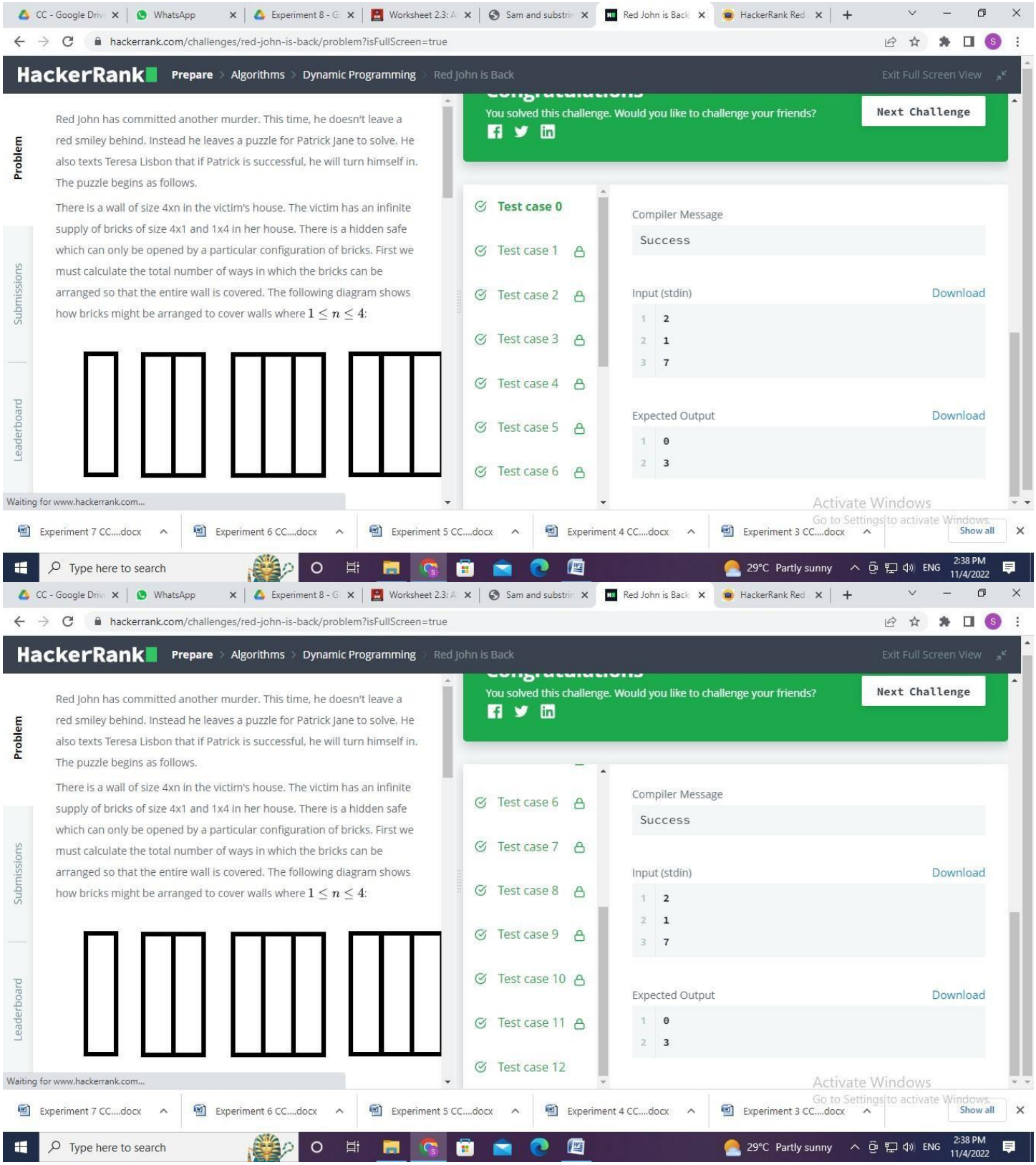
4. **Code:**

```
def 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))
```

5. Result/Output/Writing Summary:


hackerrank.com/challenges/red-john-is-back/problem?isFullScreen=true
HackerRank
Prepare > Algorithms > Dynamic Programming > Red John is Back
Exit Full Screen View
Problem
Submissions
Leaderboard
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. The puzzle begins as follows.
There is a wall of size 4xn in the victim's house. The victim has an infinite supply of bricks of size 4x1 and 1x4 in her house. There is a hidden safe which can only be opened by a particular configuration of bricks. First we must calculate the total number of ways in which the bricks can be arranged so that the entire wall is covered. The following diagram shows how bricks might be arranged to cover walls where $1 \leq n \leq 4$:
Congratulations!
You have passed the sample test cases. Click the submit button to run your code against all the test cases.
Sample Test case 0
Sample Test case 1
Input (stdin)
Download
2
1
7
Your Output (stdout)
0
3
Expected Output
Download
0
3
Activate Windows
Go to Settings to activate Windows.
Show all
Type here to search
29°C Partly sunny
ENG
2:38 PM
11/4/2022

hackerrank.com/challenges/red-john-is-back/problem?isFullScreen=true
HackerRank
Prepare > Algorithms > Dynamic Programming > Red John is Back
Exit Full Screen View
Problem
Submissions
Leaderboard
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. The puzzle begins as follows.
There is a wall of size 4xn in the victim's house. The victim has an infinite supply of bricks of size 4x1 and 1x4 in her house. There is a hidden safe which can only be opened by a particular configuration of bricks. First we must calculate the total number of ways in which the bricks can be arranged so that the entire wall is covered. The following diagram shows how bricks might be arranged to cover walls where $1 \leq n \leq 4$:
Change Theme
Language
Python 3
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))
Line: 20 Col: 48
Upload Code as File
Test against custom input
Run Code
Submit Code
Activate Windows
Go to Settings to activate Windows.
Show all
Type here to search
29°C Partly sunny
ENG
2:38 PM
11/4/2022


HackerRank
Prepare > Algorithms > Dynamic Programming > Red John is Back
Exit Full Screen View
Problem
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. The puzzle begins as follows.
There is a wall of size 4xn in the victim's house. The victim has an infinite supply of bricks of size 4x1 and 1x4 in her house. There is a hidden safe which can only be opened by a particular configuration of bricks. First we must calculate the total number of ways in which the bricks can be arranged so that the entire wall is covered. The following diagram shows how bricks might be arranged to cover walls where $1 \leq n \leq 4$:
Submissions
Leaderboard
You solved this challenge. Would you like to challenge your friends?
Next Challenge
Test case 0
Test case 1
Test case 2
Test case 3
Test case 4
Test case 5
Test case 6
Compiler Message
Success
Input (stdin)
Download
2
1
7
Expected Output
Download
0
3
Waiting for www.hackerrank.com...
Activate Windows
Go to Settings to activate Windows.
Experiment 7 CC....docx
Experiment 6 CC....docx
Experiment 5 CC....docx
Experiment 4 CC....docx
Experiment 3 CC....docx
Show all
Type here to search
29°C Partly sunny
ENG
2:38 PM
11/4/2022

HackerRank
Prepare > Algorithms > Dynamic Programming > Red John is Back
Exit Full Screen View
Problem
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. The puzzle begins as follows.
There is a wall of size 4xn in the victim's house. The victim has an infinite supply of bricks of size 4x1 and 1x4 in her house. There is a hidden safe which can only be opened by a particular configuration of bricks. First we must calculate the total number of ways in which the bricks can be arranged so that the entire wall is covered. The following diagram shows how bricks might be arranged to cover walls where $1 \leq n \leq 4$:
Submissions
Leaderboard
You solved this challenge. Would you like to challenge your friends?
Next Challenge
Test case 6
Test case 7
Test case 8
Test case 9
Test case 10
Test case 11
Test case 12
Compiler Message
Success
Input (stdin)
Download
2
1
7
Expected Output
Download
0
3
Waiting for www.hackerrank.com...
Activate Windows
Go to Settings to activate Windows.
Experiment 7 CC....docx
Experiment 6 CC....docx
Experiment 5 CC....docx
Experiment 4 CC....docx
Experiment 3 CC....docx
Show all
Type here to search
29°C Partly sunny
ENG
2:38 PM
11/4/2022

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.			