

Discover. Learn. Empower.

## **UNIVERSITY INSTITUTE OF ENGINEERING**

### **Department of Computer Science & Engineering**

**Subject Name:** Competitive Coding

**Subject Code:** 20CSP-314

**Submitted by:**

Name: Anjali Singh

UID: 20BCS9239

Section: 607

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

**Student Name:** Ruchika Raj

**UID:** 20BCS9285

**Branch:** BE CSE

**Section/Group:** 615/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:

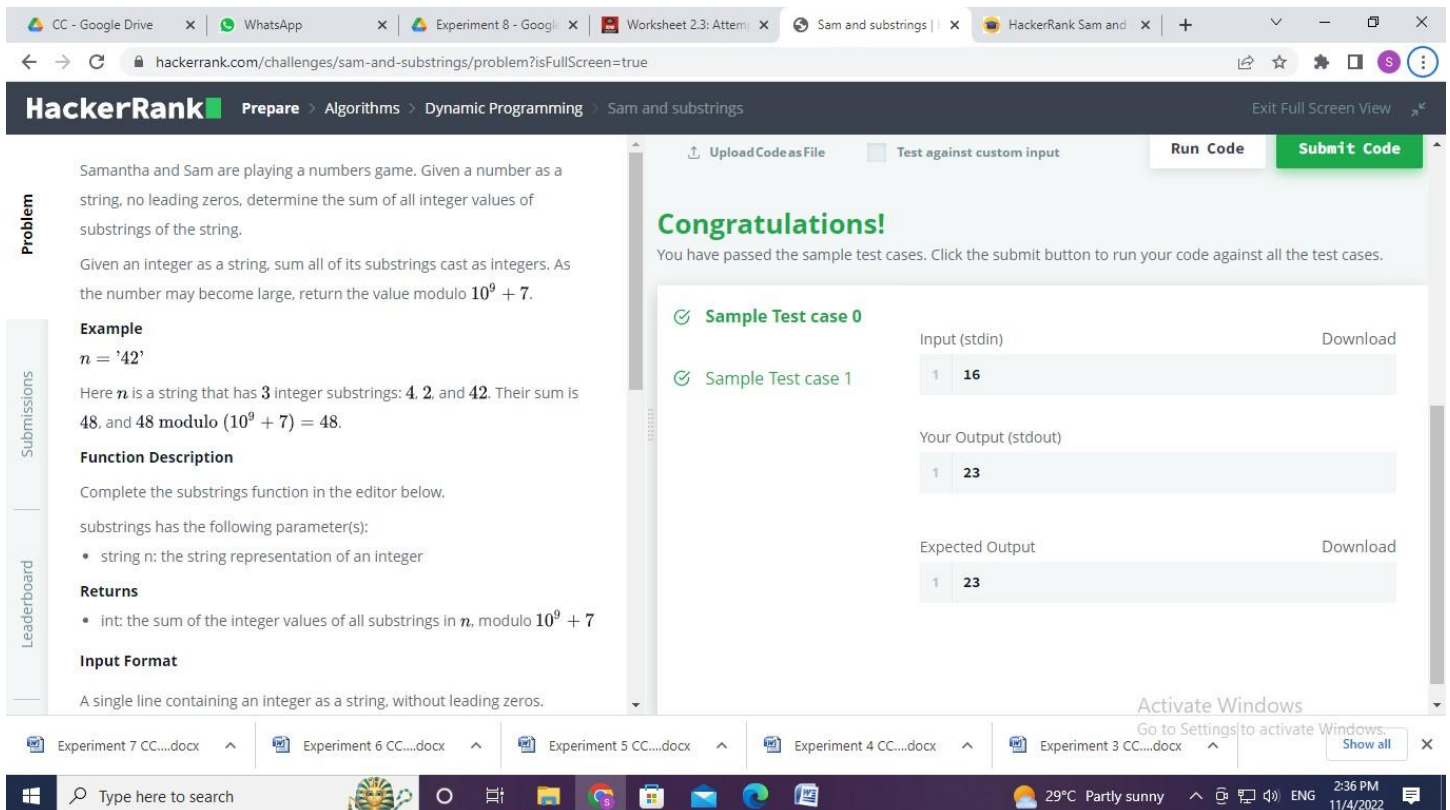
- To understand the concept of Dynamic Programming.
- To implement the concept of Dynamic Programming.
- Goal is to find the number of ways to construct an array such that consecutive positions contain different values.

### 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:



The screenshot shows the HackerRank interface for the 'Sam and Substrings' problem. The problem description on the left states: 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  $10^9 + 7$ . An example shows  $n = '42'$  resulting in a sum of 48. The function description asks to complete the `substrings` function. The submission area on the right shows two successful test cases. For Sample Test case 0, the input is '16' and the output is '23'. For Sample Test case 1, the input is '16' and the output is '23'. The 'Submit Code' button is highlighted in green.

**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**

**Test Cases**

**Sample Test case 0**

Input (stdin)

1	16
---	----

Your Output (stdout)

1	23
---	----

Expected Output

1	23
---	----

**Windows Taskbar:** Experiment 7 CC....docx, Experiment 6 CC....docx, Experiment 5 CC....docx, Experiment 4 CC....docx, Experiment 3 CC....docx, 29°C Partly sunny, 2:36 PM 11/4/2022.

**HackerRank** Prepare > Algorithms > Dynamic Programming > Sam and substrings

**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.

Change Theme Language Python 3

```

1 def solution(n):
2     s = 0
3     prev_sum = 0
4
5     for i, d in enumerate(n):
6         s_ = prev_sum * 10 + (i + 1) * int(d)
7         s += s_
8         prev_sum = s_
9     return s % (10 ** 9 + 7)
10
11 n = input()
12
13 print(solution(n))
    
```

Line: 13 Col: 19

Upload Code as File Test against custom input Run Code Submit Code

**HackerRank** Prepare > Algorithms > Dynamic Programming > Sam and substrings

**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.

**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)

16

Expected Output

23



The screenshot shows a web browser window with multiple tabs. The active tab is 'HackerRank Sam and substrings'. The page displays the 'Sam and substrings' challenge on HackerRank. The problem description states: '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  $10^9 + 7$ .' An example is given:  $n = '42'$ , where the substrings are '4', '2', and '42', with a sum of 48. The function description asks to complete the 'substrings' function. The returns section specifies an integer modulo  $10^9 + 7$ . The input format is a string  $n$ . The output format is an integer. The test cases section shows 12 test cases, all of which are passed. The compiler message shows 'Success'. The input (stdin) is '16' and the expected output is '23'. The bottom of the screenshot shows a Windows taskbar with various application icons and a system tray showing the date and time as 2:37 PM on 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)

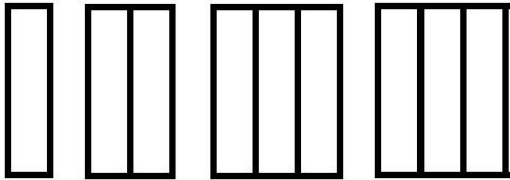
T = int(input())
for i in range(T):
    print(primes(find_configs(int(input()))+1))
```

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

**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  $4 \times n$  in the victim's house. The victim has an infinite supply of bricks of size  $4 \times 1$  and  $1 \times 4$  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**

## 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**

Input (stdin)

1	2
2	1
3	7

Download

Your Output (stdout)

1	0
2	3

Expected Output

1	0
2	3

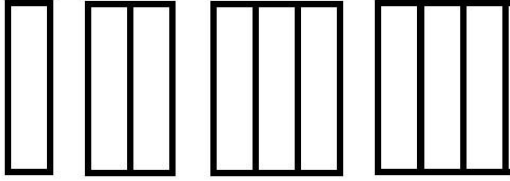
Download

Activate Windows  
Go to Settings to activate Windows.  
Show all

**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  $4 \times n$  in the victim's house. The victim has an infinite supply of bricks of size  $4 \times 1$  and  $1 \times 4$  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**

Change Theme Language Python 3

```

1 returns a list of primes < n
2
3 if n <= 2: return 0
4 sieve = [True] * n
5 for i in range(3, int(n**0.5)+1, 2):
6     if sieve[i]:
7         sieve[i::2*i] = [False]*int((n-i*i-1)/(2*i)+1)
8 return len([i for i in range(3,n,2) if sieve[i]]) + 1
9
10 def find_configs(N):
11     if N == 0:
12         return 1
13     elif N < 0:
14         return 0
15
16     return find_configs(N-1) + find_configs(N-4)
17
18 T = int(input())
19 for i in range(T):
20     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



CC - Google Drive x WhatsApp x Experiment 8 - G x Worksheet 2.3: A x Sam and substrin x Red John is Back x HackerRank Red x +

hackerrank.com/challenges/red-john-is-back/problem?isFullScreen=true

## HackerRank

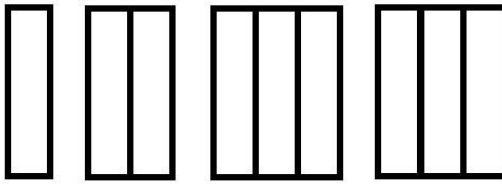
Prepare > Algorithms > Dynamic Programming > Red John is Back

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  $4 \times n$  in the victim's house. The victim has an infinite supply of bricks of size  $4 \times 1$  and  $1 \times 4$  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  $1 \leq n \leq 4$ :



Waiting for www.hackerrank.com...

**Submissions**

**Leaderboard**

**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](#)

```
1 2
2 1
3 7
```

**Expected Output** [Download](#)

```
1 0
2 3
```

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

Type here to search

CC - Google Drive x WhatsApp x Experiment 8 - G x Worksheet 2.3: A x Sam and substrin x Red John is Back x HackerRank Red x +

hackerrank.com/challenges/red-john-is-back/problem?isFullScreen=true

## HackerRank

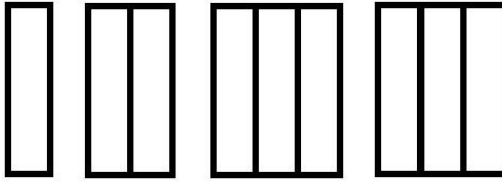
Prepare > Algorithms > Dynamic Programming > Red John is Back

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  $4 \times n$  in the victim's house. The victim has an infinite supply of bricks of size  $4 \times 1$  and  $1 \times 4$  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  $1 \leq n \leq 4$ :



Waiting for www.hackerrank.com...

**Submissions**

**Leaderboard**

**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](#)

```
1 2
2 1
3 7
```

**Expected Output** [Download](#)

```
1 0
2 3
```

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

Type here to search

29°C Partly sunny 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.			