



## Experiment 8

### Dynamic programming

**Student Name:** Bhanu Pundir

**UID:** 20BCS1439

**Branch:** BE CSE

**Section/Group:** 62-B

**Semester:** 5th

**Date of Performance:** 04 Nov 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:

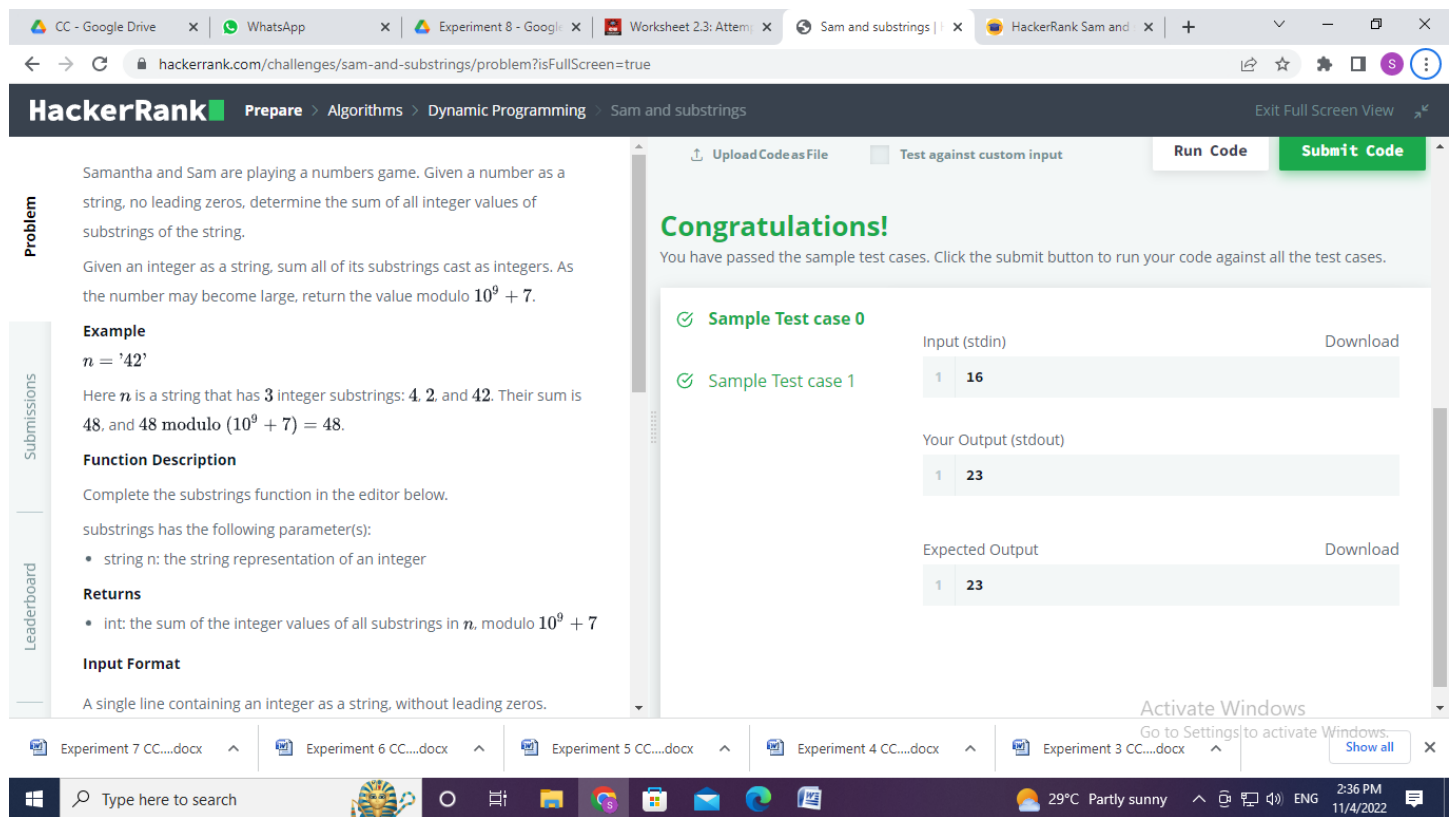
```
def solution(n):  
    s = 0  
    prev_sum = 0  
  
    for i, d in enumerate(n):
```

```
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 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  $10^9 + 7$ .

**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.

The submission section shows two sample test cases:

Sample Test case	Input (stdin)	Your Output (stdout)	Expected Output
Sample Test case 0	1	16	16
Sample Test case 1	1	23	23

The page displays a 'Congratulations!' message: 'You have passed the sample test cases. Click the submit button to run your code against all the test cases.'



# DEPARTMENT OF ACADEMIC AFFAIRS

Discover. Learn. Empower.

NAAC  
GRADE A+  
ACCREDITED UNIVERSITY

CC - Google Drive

WhatsApp

Experiment 8 - Google

Worksheet 2.3: Atten

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

Submissions

Leaderboard

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

Activate Windows  
Go to Settings to activate Windows.  
[Show all](#)

Experiment 7 CC....docx

Experiment 6 CC....docx

Experiment 5 CC....docx

Experiment 4 CC....docx

Experiment 3 CC....docx

Go to Settings to activate Windows.

Type here to search

29°C Partly sunny

ENG

2:36 PM 11/4/2022



# DEPARTMENT OF ACADEMIC AFFAIRS

Discover. Learn. Empower.



CC - Google Drive x WhatsApp x Experiment 8 - Google x Worksheet 2.3: Atten x Sam and substrings | x HackerRank Sam and x

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**

Waiting for heapanalytics.com...

### Congratulations

You solved this challenge. Would you like to challenge your friends?

[Next Challenge](#)

[Facebook](#) [Twitter](#) [LinkedIn](#)

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. [Show all](#)

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 2:37 PM 11/4/2022

The screenshot shows a web browser window displaying the HackerRank challenge 'Sam and substrings'. The browser's address bar shows the URL: [hackerrank.com/challenges/sam-and-substrings/problem?isFullScreen=true](https://hackerrank.com/challenges/sam-and-substrings/problem?isFullScreen=true). The page has a dark header with the HackerRank logo and navigation links: 'Prepare', 'Algorithms', 'Dynamic Programming', and 'Sam and substrings'. On the left, a sidebar contains 'Problem', 'Submissions', and 'Leaderboard' tabs. The main content area is divided into two columns. The left column contains the problem description: '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  $10^9 + 7$ .' It also includes an example with  $n = '42'$  and a function description for the 'substrings' function. The right column shows a 'Congratulations' message, a list of 12 test cases all marked as 'Success', and a 'Compiler Message' section showing 'Success'. Below the test cases, there are sections for 'Input (stdin)' and 'Expected Output', both showing a single test case with input '16' and output '23'. At the bottom of the browser window, a Windows taskbar is visible with several open documents 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:

#### Dynamic Programming

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:

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

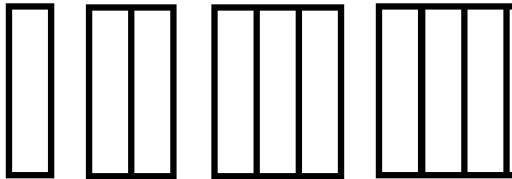
hackerank.com/challenges/red-john-is-back/problem?isFullScreen=true

**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  $1 \leq n \leq 4$ :



**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) Download

1	2
2	1
3	7

✓ Sample Test case 1

Your Output (stdout)

1	0
2	3

Expected Output Download

1	0
2	3

Activate Windows  
Go to Settings to activate Windows. Show all

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





# DEPARTMENT OF ACADEMIC AFFAIRS

Discover. Learn. Empower.



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

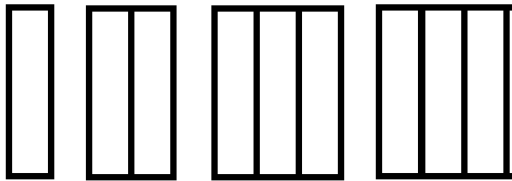
Prepare > Algorithms > Dynamic Programming > Red John is Back

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  $1 \leq n \leq 4$ :



### Submissions

### Leaderboard

```
def primes(n):  
    return a list of primes < n  
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*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

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





# DEPARTMENT OF ACADEMIC AFFAIRS

Discover. Learn. Empower.

**NAAC  
GRADE A+**  
ACCREDITED UNIVERSITY

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  $1 \leq n \leq 4$ :

Submissions

Leaderboard

Waiting for www.hackerrank.com...

### Test cases

- 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)

1	2
2	1
3	7

Download

### Expected Output

1	0
2	3

Download

Activate Windows

Go to Settings to activate Windows. Show all

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

**HackerRank** Prepare > Algorithms > Dynamic Programming > Red John is Back

**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  $1 \times 1$  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

Waiting for www.hackerrank.com...

**Test Cases**

- 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)**

```
2
1
7
```

**Expected Output**

```
0
3
```

Activate Windows

## Learning outcomes (What I have learnt):

- Learned the concept of Dynamic Programming.
- Learned 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.			