

## **Experiment 8 (Dynamic Programming)**

**Student Name: Lipakshi**

**UID: 20BCS5082**

**Branch: BE CSE**

**Section/Group: 607 / B**

**Semester: 5<sup>th</sup>**

**Date of performance: 02.11.22**

**Subject: Competitive Coding**

**Subject Code: 20CSP\_314**

### **1. Aim/Overview of the Practical:**

- a. Equal.
- b. Sam and substrings.

### **2. Task to be done / Which logistics used:**

- a. Christy is interning at Hacker Rank. One day she has to distribute some chocolates to her colleagues. She is biased towards her friends and plans to give them more than the others. One of the program managers hears of this and tells her to make sure everyone gets the same number.

To make things difficult, she must equalize the number of chocolates in a series of operations. For each operation, she can give 1, 2 and 5 pieces to all but one colleague. Everyone who gets a piece in a round receives the same number of pieces.

Given a starting distribution, calculate the minimum number of operations needed so that every colleague has the same number of pieces.

**Example**

**arr=[1,1,5]**

arr represents the starting numbers of pieces for each colleague. She can give 2 pieces to the first two and the distribution is then [3,3,5]. On the next round, she gives the same two 2 pieces each, and everyone has the same number: [5,5,5]. Return the number of rounds, 2.

- b. 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  $10^9+7$ .

### 3. Steps for experiment/practical/Code:

#### a. Equal

```
int main()
{
    int t,i,j,n,a[100000],min,ans,curr;
    scanf("%d",&t);
    while(t--)
    {
        scanf("%d",&n);min=100000;
        for (int i = 0; i < n; i++)
        {
            scanf("%d",&a[i]);
            if(a[i]<min)min=a[i];
        }
        ans=0;
        for(i=0;i<n;i++)
        {
            j=a[i]-min;
            ans=ans+j/5;
            ans=ans+(j%5)/2;
            if((j%5)%2)ans++;
            ;
        }
        curr=ans;
        ans=0;min=min-1;
```

```

for(i=0;i<n;i++)
{
    j=a[i]-min;
    ans=ans+j/5;
    ans=ans+(j%5)/2;
    if((j%5)%2)ans++;
    ;
}
if(ans<curr){curr=ans;}
ans=0;min=min-1;
for(i=0;i<n;i++)
{
    j=a[i]-min;
    ans=ans+j/5;
    ans=ans+(j%5)/2;
    if((j%5)%2)ans++;
    ;
}
if(ans<curr)curr=ans
; ans=0;min=min-1;
for(i=0;i<n;i++)
{
    j=a[i]-min;
    ans=ans+j/5;
    ans=ans+(j%5)/2;
    if((j%5)%2)ans++;
    ;
}

```



**DEPARTMENT OF**  
**COMPUTER SCIENCE &**  
**ENGINEERING**

Discover. Learn. Empower.

```
if(ans<curr)curr=ans;
```

```

        ans=0;min=min-1;
        for(i=0;i<n;i++)
        {
            j=a[i]-min;
            ans=ans+j/5;
            ans=ans+(j%5)/2;
            if((j%5)%2)ans++
            ;
        }
        if(ans<curr)curr=ans;

        printf("%d \n",curr);
    }
}

```

### b. Sam and Substring:

```

int main()
{
    ios_base::sync_with_stdio(0);
    string s;
    cin >> s;

    ll count_before = 0, result_before = 0, result = 0;
    for(size_t i = 0; i < s.size(); i++)
    {
        ll digit = (s[i] - '0');
        ++count_before;
        result_before = (10 * result_before + count_before * digit) % MODULO;
    }
}

```

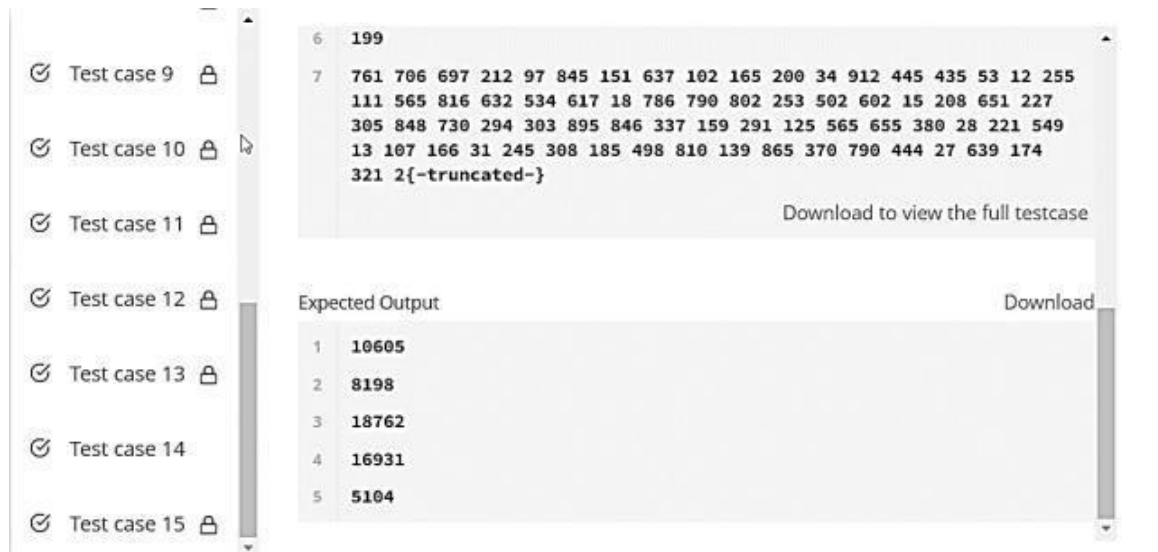
```

    result = (result + result_before) % MODULO;
  }
  cout << result <<
  endl; return 0;
}

```

## Result/Output/Writing Summary:

### a. Equal:



The screenshot shows a list of test cases on the left and the details for 'Test case 9' on the right.

**Test Cases List:**

- Test case 9
- Test case 10
- Test case 11
- Test case 12
- Test case 13
- Test case 14
- Test case 15

**Test case 9 Details:**

6 199

7 761 706 697 212 97 845 151 637 102 165 200 34 912 445 435 53 12 255  
 111 565 816 632 534 617 18 786 790 802 253 502 602 15 208 651 227  
 305 848 730 294 303 895 846 337 159 291 125 565 655 380 28 221 549  
 13 107 166 31 245 308 185 498 810 139 865 370 790 444 27 639 174  
 321 2{-truncated-}

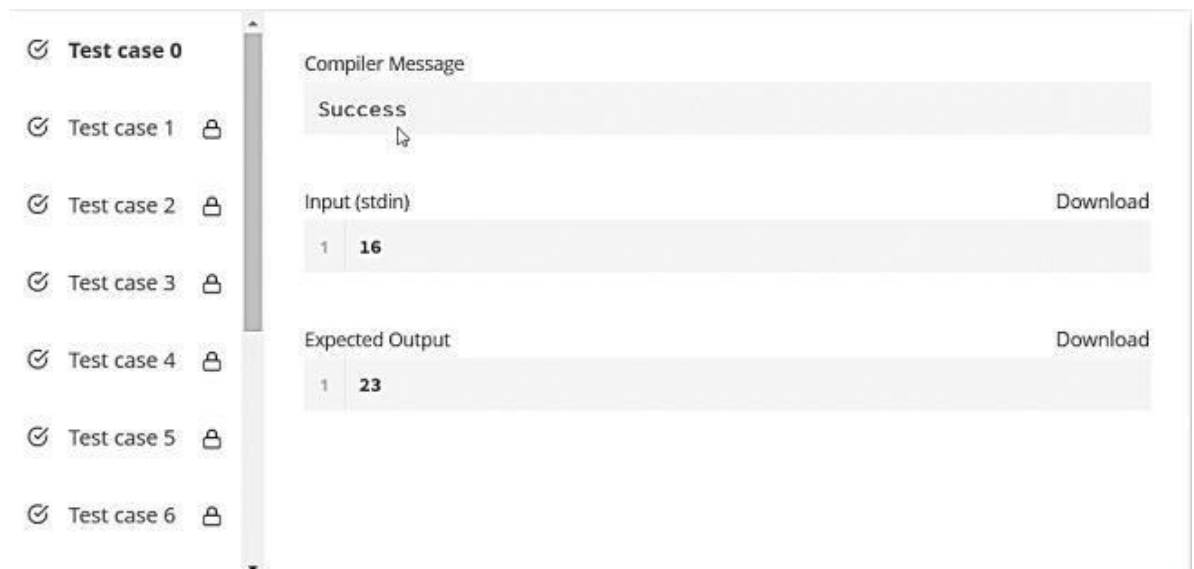
[Download to view the full testcase](#)

**Expected Output:**

1 10605  
 2 8198  
 3 18762  
 4 16931  
 5 5104

[Download](#)

### b. Sam and Substring:



The screenshot shows a list of test cases on the left and the details for 'Test case 0' on the right.

**Test Cases List:**

- Test case 0
- Test case 1
- Test case 2
- Test case 3
- Test case 4
- Test case 5
- Test case 6

**Test case 0 Details:**

**Compiler Message:**

Success

**Input (stdin):**

1 16

[Download](#)

**Expected Output:**

1 23

[Download](#)

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

- a. Learnt about dynamic programming.
- b. Got an overview of the implementation of strings.
- c. Get to know about crucial test cases.