

**Velagapudi Ramakrishna Siddhartha Engineering College**

**ADVANCE PROGRAMMING LAB – 2**

**Code : 20IT5352**



## WEEK – 1

**Aim :** Vlad enjoys listening to music. He lives in Sam's Town. A few days ago he had a birthday, so his parents gave him a gift: MP3-player! Vlad was the happiest man in the world! Now he can listen his favorite songs whenever he wants!

Vlad built up his own playlist. The playlist consists of  $N$  songs, each has a unique positive integer length. Vlad likes all the songs from his playlist, but there is a song, which he likes more than the others. It's named "Uncle Johny".

After creation of the playlist, Vlad decided to sort the songs in increasing order of their lengths. For example, if the lengths of the songs in playlist was  $\{1, 3, 5, 2, 4\}$  after sorting it becomes  $\{1, 2, 3, 4, 5\}$ . Before the sorting, "Uncle Johny" was on  $K$ -th position (1-indexing is assumed for the playlist) in the playlist.

Vlad needs your help! He gives you all the information of his playlist. Your task is to find the position of "Uncle Johny" in the sorted playlist.

### **Sample Input :**

The first line of the input contains an integer  $T$  denoting the number of test cases. The description of  $T$  test cases follows.

The first line of each test case contains one integer  $N$  denoting the number of songs in Vlad's playlist. The second line contains  $N$  space-separated integers  $A_1, A_2, \dots, A_N$  denoting the lengths of Vlad's songs. The third line contains the only integer  $K$  - the position of "Uncle Johny" in the initial playlist.

### **Sample Output :**

For each test case, output a single line containing the position of "Uncle Johny" in the sorted playlist.

### **Program :**

```
T = int(input())
```

```
while T > 0:
```

```

il = list(map(int, input().split()))

N = il[0]

M = il[-1]

if M >= N:

    print("NO")

else:

    if (N%M == 0) and (((N//M)*M) == N):

        print("YES")

    else:

        print("NO")

```

$T = T - 1$

## Output :

The screenshot shows a problem titled "Uncle Johny" on the CodeChef website. The problem statement is as follows:

**Problem**  
Read problems statements in [Mandarin Chinese](#) and [Russian](#).  
Vlad enjoys listening to music. He lives in Sam's Town. A few days ago he had a birthday, so his parents gave him a gift: MP3-player! Vlad was the happiest man in the world! Now he can listen his favorite songs whenever he wants!  
Vlad built up his own playlist. The playlist consists of **N** songs, each has a **unique** positive integer length. Vlad likes all the songs from his playlist, but there is a song, which he likes more than the others. It's named "Uncle Johny".  
After creation of the playlist, Vlad decided to sort the songs in increasing order of their lengths. For example, if the lengths of the songs in playlist was {1, 3, 5, 2, 4} after sorting it becomes {1, 2, 3, 4, 5}. Before the sorting, "Uncle Johny" was on **K**-th position (1-indexing is assumed for the playlist) in the

The code submitted is:

```

1 T = int(input())
2
3 while T > 0:
4     il = list(map(int, input().split()))
5     N = il[0]
6     M = il[-1]
7
8     if M >= N:
9         print("NO")
10    else:
11        if (N*M == 0) and (((N//M)*M) == N):
12            print("YES")
13        else:
14            print("NO")
15
16    T = T - 1
17
18

```

The code is run under the PYTH 3 environment. The status bar indicates the time as 15:13.

Date : .....

SHEET No: .....

The screenshot shows a programming contest interface. At the top, there are tabs for Statement, Hints, Submissions, Solution, and Ask a Doubt. A dropdown menu shows PYTH 3 selected. On the right, there are icons for download, copy, paste, settings, and help. Below the tabs, a section titled "Problem" contains the problem statement in English, which is identical to the one provided in the question. It includes a story about Vlad and his playlist, constraints (N unique positive integers), and a note about sorting. There are also sections for Mandarin Chinese and Russian translations. The "Test against Custom Input" section shows test cases with inputs 5, 5, 1 2 3 9 4, and 1. Below this, there are buttons for "Upload code as file" (with a file icon), "Compile & Run" (with a play icon), and "Submit Code" (blue button). A status bar at the bottom indicates "Status: ✓ Correct Answer" and "Submission ID: 73055930". It also shows "Time: 0.05s". To the right, a message says "Congratulations on solving the problem. Visit our practice section to solve more interesting problems" and a link "View another problem".

**Time Complexity :**  $O(n)$

**Space Compexity :**  $O(n)$

**Result :** Sucessfully Executed the Program.

## WEEK – 2

**Aim :** The game of billiards involves two players knocking 3 balls around on a green baize table. Well, there is more to it, but for our purposes this is sufficient.

The game consists of several rounds and in each round both players obtain a score, based on how well they played. Once all the rounds have been played, the total score of each player is determined by adding up the scores in all the rounds and the player with the higher total score is declared the winner.

The Siruseri Sports Club organises an annual billiards game where the top two players of Siruseri play against each other. The Manager of Siruseri Sports Club decided to add his own twist to the game by changing the rules for determining the winner. In his version, at the end of each round, the cumulative score for each player is calculated, and the leader and her current lead are found. Once all the rounds are over the player who had the maximum lead at the end of any round in the game is declared the winner.

### **Sample Input :**

The first line of the input will contain a single integer  $N$  ( $N \leq 10000$ ) indicating the number of rounds in the game. Lines 2,3,..., $N+1$  describe the scores of the two players in the  $N$  rounds. Line  $i+1$  contains two integer  $S_i$  and  $T_i$ , the scores of the Player 1 and 2 respectively, in round  $i$ . You may assume that  $1 \leq S_i \leq 1000$  and  $1 \leq T_i \leq 1000$ .

### **Sample Output :**

Your output must consist of a single line containing two integers  $W$  and  $L$ , where  $W$  is 1 or 2 and indicates the winner and  $L$  is the maximum lead attained by the winner.

**Program :**

```
import numpy as np

t = int(input())

temp = t

players = []

player1 = []

player2 = []

while t > 0:

    score = list(map(int, input().split()[:2]))

    player1.append(score[0])

    player2.append(score[-1])

    players.append(score)

    t = t - 1

dict_players = {}

dict_players[1] = player1

dict_players[2] = player2

p1 = dict_players[1]

p2 = dict_players[2]

p1 = np.array(p1)

p2 = np.array(p2)

player_leads = list(abs(p1-p2))

leader_round = {}

for i in range(1, temp+1):

    if p1[i-1] > p2[i-1]:
```

```

value = player_leads[i-1]
leader_round[value] = 'p1'

else:

value = player_leads[i-1]
leader_round[value] = 'p2'

final_result = sorted(leader_round)

maxp = max(final_result)

if leader_round[maxp] == 'p1':
    print(1, maxp)

elif leader_round[maxp] == 'p2':
    print(2, maxp)

```

## Output :

The screenshot shows the CodeChef platform interface for the 'The Lead Game' problem. The problem statement, input, and output sections are visible on the left, while the code editor and test results are on the right.

**Problem Statement:**

The Lead Game

Problem Code: TLG ✓

Difficulty Rating: 790 Show Tags

**Code Editor (PYTH 3):**

```

32
33     leader_round = {}
34     for i in range(1, temp+1):
35         if p1[i-1] > p2[i-1]:
36             value = player_leads[i-1]
37             leader_round[value] = 'p1'
38         else:
39             value = player_leads[i-1]
40             leader_round[value] = 'p2'.
41
42     final_result = sorted(leader_round)
43
44     #print("Leader dictionary : ", leader_round)
45     #print("Final result dictionary : ", final_result)
46
47     maxp = max(final_result)
48     if leader_round[maxp] == 'p1':
49         print(1, maxp)
50     elif leader_round[maxp] == 'p2':
51         print(2, maxp)
52

```

**Test Results:**

Sample 1: Test against Custom Input

51:18

**Statement**    **Hints**    **Submissions**    **Solution**    **Ask a Doubt**

PYTHON

```

15:     else:
16:         a.append(r-s)
17:         b.append(2)
18:     print(b[a.index(max(a))],max(a))

```

17:32

**Test against Custom Input**

```

5
140 82
89 134
90 110

```

Upload code as file        

Status: ✓ Correct Answer    Submission ID: 73056243

Time: 0.39s

Congratulations on solving the problem. Visit our practice section to solve more interesting problems    [View another problem →](#)

Up ↑

**Time Complexity :**  $O(2n)$

**Space Complexity :**  $O(n)$

**Result :** Sucessfully Executed the program.

## WEEK – 3

**Aim :** Devu has  $n$  weird friends. It's his birthday today, so they thought that this is the best occasion for testing their friendship with him. They put up conditions before Devu that they will break the friendship unless he gives them a grand party on their chosen day. Formally,  $i^{th}$  friend will break his friendship if he does not receive a grand party on  $d_i^{th}$  day.

Devu despite being as rich as Gatsby, is quite frugal and can give at most one grand party daily. Also, he wants to invite only one person in a party. So he just wonders what is the maximum number of friendships he can save. Please help Devu in this tough task !!

**Sample Input :**

- The first line of the input contains an integer  $T$  denoting the number of test cases. The description of  $T$  test cases follows.
- First line will contain a single integer denoting  $n$ .
- Second line will contain  $n$  space separated integers where  $i^{th}$  integer corresponds to the day  $d_i^{th}$  as given in the problem.

**Sample Output :**

Print a single line corresponding to the answer of the problem.

**Program :**

```
test_cases = int(input())
while test_cases > 0:
    no_of_friends = int(input())
    friends_list = list(map(int, input().split()[:no_of_friends]))
    friends_set = set(friends_list)
```

```

if friends_list == friends_set:
    print(len(friends_list))
else:
    print(len(friends_set))
test_cases -= test_cases

```

## Output :

The screenshot shows a CodeChef problem page for 'Devu and friendship testing'. The problem statement is as follows:

**Problem**  
Read problems statements in [Mandarin Chinese](#) and [Russian](#) as well.  
Devu has  $n$  weird friends. Its his birthday today, so they thought that this is the best occasion for testing their friendship with him. They put up conditions before Devu that they will break the friendship unless he gives them a grand party on their chosen day. Formally,  $i^{th}$  friend will break his friendship if he does not receive a grand party on  $d_i^{th}$  day.

Devu despite being as rich as Gatsby, is quite frugal and can give at most one grand party daily. Also, he wants to invite only one person in a party. So he just wonders what is the maximum number of friendships he can save. Please help Devu in this tough task !!

The solution code is:

```

1 test_cases = int(input())
2 while test_cases > 0:
3     no_of_friends = int(input())
4     friends_list = list(map(int, input().split()[:no_of_friends]))
5
6     friends_set = set(friends_list)
7     if friends_list == friends_set:
8         print(len(friends_list))
9     else:
10        print(len(friends_set))
11
12     test_cases -= test_cases|
```

The code editor interface shows the language selected as PYTH 3. The status bar indicates the time as 12:28.

The screenshot shows a programming problem interface. At the top, there are tabs for Statement, Hints, Submissions, Solution, Ask a Doubt, and a dropdown for PYTH 3. On the right, there are icons for download, copy, and settings. Below the tabs, the word "Problem" is highlighted.

**Problem**

Read problems statements in [Mandarin Chinese](#) and [Russian](#) as well.

Devu has  $n$  weird friends. It's his birthday today, so they thought that this is the best occasion for testing their friendship with him. They put up conditions before Devu that they will break the friendship unless he gives them a grand party on their chosen day. Formally,  $i^{\text{th}}$  friend will break his friendship if he does not receive a grand party on  $d_i^{\text{th}}$  day.

Devu despite being as rich as Gatsby, is quite frugal and can give at most one grand party daily. Also, he wants to invite only one person in a party. So he just wonders what is the maximum number of friendships he can save. Please help Devu in this tough task !!

**Input**

- The first line of the input contains an integer  $T$  denoting the number of test cases. The description of  $T$  test cases follows.
- First line will contain a single integer denoting  $n$ .
- Second line will contain  $n$  space separated integers where  $i^{\text{th}}$  integer corresponds to the day  $d_i^{\text{th}}$  as given in the problem.

**Output**

Print a single line corresponding to the answer of the problem.

**Constraints**

Test against Custom Input

2  
2  
3 2  
2

Upload code as file  Compile & Run  Submit Code

Status: ✓ Correct Answer Submission ID: 73056928

Time:  
0.14s

Congratulations on solving the problem. Visit our practice section to solve more interesting problems [View another problem](#)

**Time Complexity :**  $O(n)$

**Space Complexity :**  $O(n)$

**Result :** Sucessfully Executed the Program.

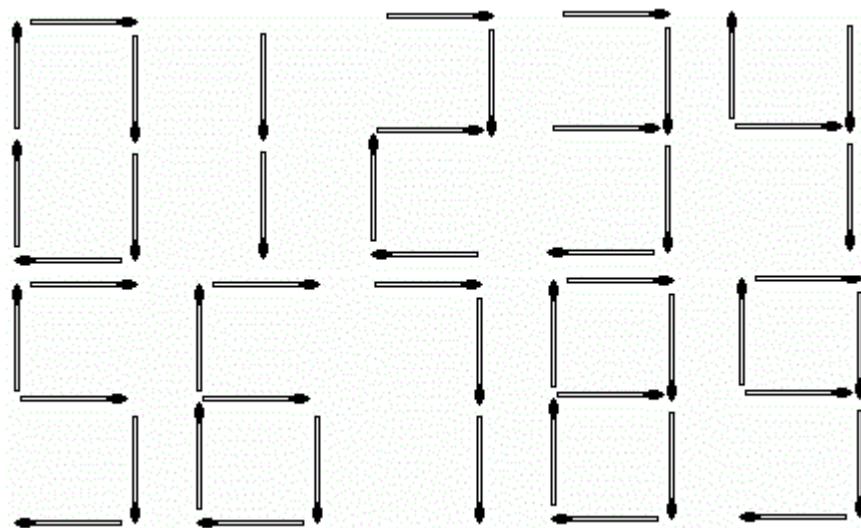
## WEEK – 4

### Task – 1:

**Aim :** Chef's son Chefu found some matches in the kitchen and he immediately starting playing with them.

The first thing Chefu wanted to do was to calculate the result of his homework — the sum of AA and BB, and write it using matches. Help Chefu and tell him the number of matches needed to write the result.

Digits are formed using matches in the following way:



#### Sample Input :

- The first line of the input contains a single integer  $T$  denoting the number of test cases. The description of  $T$  test cases follows.
- The first and only line of each test case contains two space-separated integers  $AA$  and  $BB$ .

#### Sample Output :

For each test case, print a single line containing one integer — the number of matches needed to write the result ( $A+BA+B$ ).

**Program :**

```
test_cases = int(input( ))  
for i in range(test_cases):  
    match = list(map(int, input( ).split( )))[:2]  
    mat_res = match[0] + match[1]  
  
    number_length = [6,2,5,5,4,5,6,3,7,6]  
    res_str = str(mat_res)  
    len_res = len(res_str)  
  
    min_count = 0  
    for i in range(len_res):  
        x = int(res_str[i])  
        y = number_length[x]  
        min_count = min_count + int(y)  
  
    print(min_count)
```

**Time Complexity :**  $O(n^2)$

**Space Complexity :**  $O(n)$

## Output :

Screenshot of a browser window showing the CodeChef platform for solving the "Playing with Matches" problem.

**Header:** New Tab | Adp lab-2 - Google Docs | Playing with Matches - Submit | +

**Address Bar:** https://www.codechef.com/submit/MATCHES?tab=statement

**User Information:** Upgrade To Pro | msai\_45 | Logout

**Navigation:** Back to Practice | Switch to Old Problems Page | Prev Problem | Next Problem

**Problem Details:**

- Problem Code: MATCHES | Contest Code: COOK110
- Difficulty Rating: 986 | Show Tags

**Statement:**

**Code Editor (PYTH 3):**

```

1 # cook your dish here
2 test_cases = int(input())
3 for i in range(test_cases):
4     match = list(map(int, input().split()))[:2]
5     mat_res = match[0] + match[1]
6
7     number_length = [6,2,5,4,5,6,3,7,6]
8     res_str = str(mat_res)
9     len_res = len(res_str)
10
11    min_count = 0
12    for i in range(len_res):
13        x = int(res_str[i])
14        y = number_length[x]
15        min_count = min_count + int(y)
16
17    print(min_count)

```

**Test against Custom Input:**

```

3
123 234
10101 1010
4 4

```

**Execution Results:**

- Status: ✓ Correct Answer | Submission ID: Z4500768
- Time: 0.03s

**Explanation:**

**Sample 1:**

Input	Output
3 123 234 10101 1010 4 4	13 10 7

**Conclusion:**

Congratulations on solving the problem. Visit our practice section to solve more interesting problems | View another problem

07:07 PM 19-09-2022

**Result :** Sucessfully Executed the Program.

## Task – 2

**Aim :** You are given a binary string AA of length NN.

You can perform the following type of operation on the string A:

- Choose two different indices i and j ( $1 \leq i, j \leq N$ );
- Change  $A_i$  and  $A_j$  to  $A_i \oplus A_j$ . Here  $\oplus$  represents the bitwise XOR operation.

Find the minimum number of operations required to convert the given string into a palindrome.

### Sample Input :

- First line of the input contains TT, the number of test cases. Then the test cases follow.
- First line of each test case contains an integer NN denoting the size of the string.
- Second line of each test case contains a binary string AA of length NN containing 00s and 11s only.

### Sample Output :

For each test case, print the **minimum** number of operations required to make the string a palindrome.

### Program :

```
import math

for _ in range(int(input())):
    length = int(input())
    string = input()
    h = length//2
    if length%2 == 0:
```

```

half_1 = string[:h]
half_2 = string[h:]
half_2 = half_2[::-1]

else:
    half_1 = string[:h]
    half_2 = string[h+1:]
    half_2 = half_2[::-1]

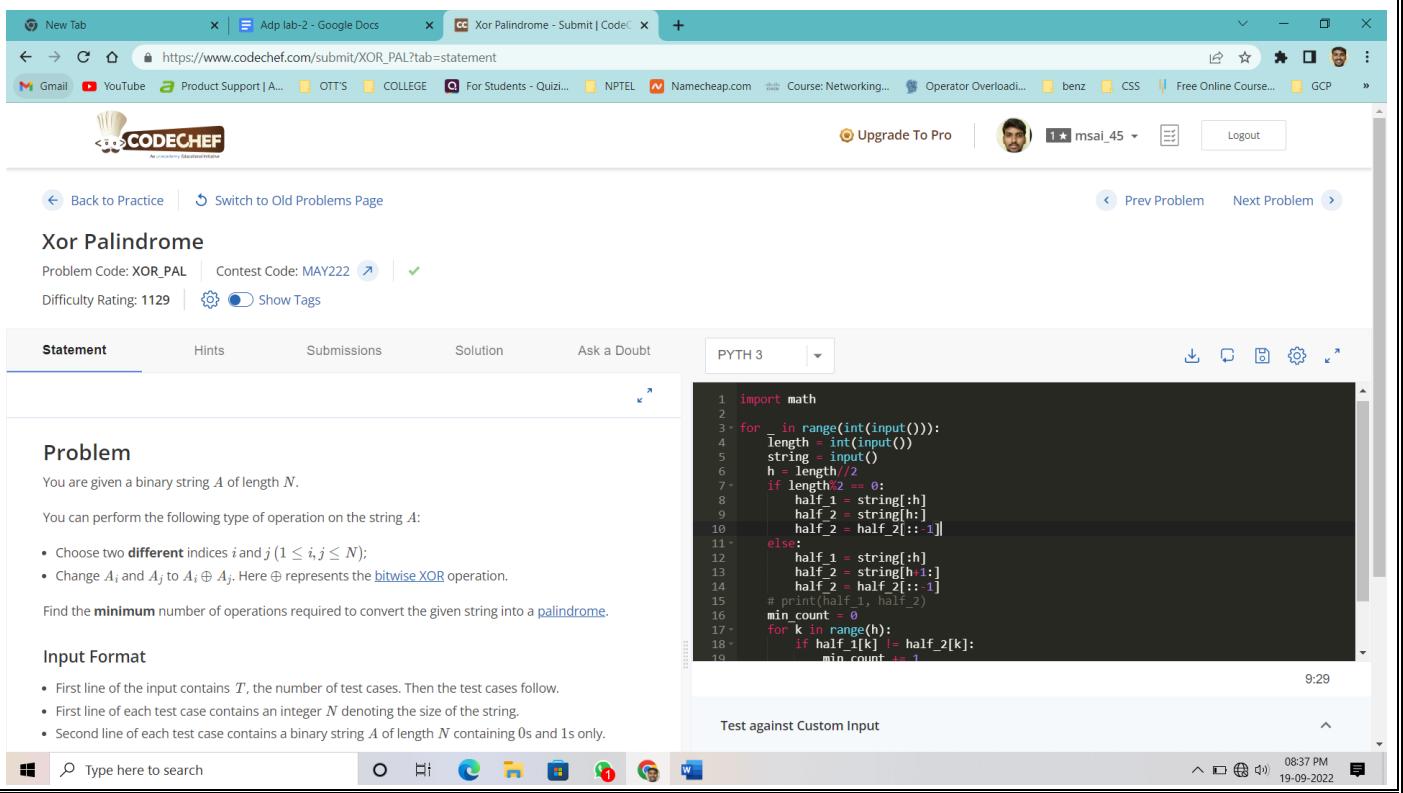
min_count = 0

for k in range(h):
    if half_1[k] != half_2[k]:
        min_count += 1

print(math.ceil(min_count/2))

```

## Output :



The screenshot shows a browser window with the following details:

- Address Bar:** https://www.codechef.com/submit/XOR\_PAL?tab=statement
- Header:** New Tab, Adp lab-2 - Google Docs, Xor Palindrome - Submit | CodeChef
- User Information:** msai\_45, Logout
- CodeChef Logo:** A chef's hat icon.
- Navigation:** Back to Practice, Switch to Old Problems Page, Prev Problem, Next Problem
- Problem Details:**
  - Name:** Xor Palindrome
  - Problem Code:** XOR\_PAL
  - Contest Code:** MAY22
  - Difficulty Rating:** 1129
  - Show Tags:** button
- Statement:** A tab labeled PYTH 3 contains the following Python code:

```

1 import math
2
3 for _ in range(int(input())):
4     length = int(input())
5     string = input()
6     h = length//2
7     if length%2 == 0:
8         half_1 = string[:h]
9         half_2 = string[h:]
10        half_2 = half_2[::-1]
11    else:
12        half_1 = string[:h]
13        half_2 = string[h+1:]
14        half_2 = half_2[::-1]
15    # print(half_1, half_2)
16    min_count = 0
17    for k in range(h):
18        if half_1[k] != half_2[k]:
19            min_count += 1

```

- Input Format:**
  - First line of the input contains  $T$ , the number of test cases. Then the test cases follow.
  - First line of each test case contains an integer  $N$  denoting the size of the string.
  - Second line of each test case contains a binary string  $A$  of length  $N$  containing 0s and 1s only.
- Output:** The code prints the minimum number of operations required to convert the given string into a palindrome.
- Environment:** The code is written in Python 3. The browser interface includes a search bar, taskbar icons, and a status bar showing the date and time.

The screenshot shows a web browser window with multiple tabs open. The active tab is for a CodeChef challenge titled "Xor Palindrome - Submit". The page includes sections for "Statement", "Hints", "Submissions", "Solution", and "Ask a Doubt". It also shows a file upload area for Python code, with "PYTH 3" selected. A "Compile & Run" button and a "Submit Code" button are visible.

**Input Format**

- First line of the input contains  $T$ , the number of test cases. Then the test cases follow.
- First line of each test case contains an integer  $N$  denoting the size of the string.
- Second line of each test case contains a binary string  $A$  of length  $N$  containing 0s and 1s only.

**Output Format**

For each test case, print the **minimum** number of operations required to make the string a palindrome.

**Constraints**

- $1 \leq T \leq 1000$
- $1 \leq N \leq 2 \cdot 10^5$
- Sum of  $N$  over all test cases does not exceed  $2 \cdot 10^5$ .

**Sample 1:**

Input	Output
4	0
5	1
11011	0
7	1
0111010	
1	
1	

**Status:** ✓ Correct Answer  
Submission ID: Z4506895

Time: 0.04s

Sub-Task	Task #	Result (time)
1	2	AC (0.040263)
1	3	AC (0.036041)
1	4	AC (0.036311)
1	5	AC (0.036680)
1	6	AC (0.036733)

Subtask Score: 100.00%      Result - AC

Total Score = 100.00%

Windows taskbar at the bottom: Type here to search, File Explorer, Task View, Start button, Taskbar icons, Date/Time (19-09-2022), 08:38 PM.

**Time Complexity :**  $O(n^2)$

**Space Complexity :**  $O(n)$

**Result :** Sucessfully Executed the program.

## Task – 3

**Aim :** Chef likes to play table tennis. He found some statistics of matches which described who won the points in order. A game shall be won by the player first scoring 11 points except in the case when both players have 10 points each, then the game shall be won by the first player subsequently gaining a lead of 2 points. Could you please help the Chef find out who the winner was from the given statistics? (It is guaranteed that statistics represent always a valid, finished match.)

### Sample Input :

The first line of the input contains an integer **T**, denoting the number of test cases. The description of **T** test cases follows. Each test case consist a binary string **S**, which describes a match. '0' means Chef lose a point, whereas '1' means he won the point.

### Sample Output :

For each test case, output on a separate line a string describing who won the match. If Chef won then print "WIN" (without quotes), otherwise print "LOSE" (without quotes).

### Program :

```
chef = int(input())
for _ in range(chef):
    score = input()
    score = list(score)
    one = score.count('1')
    zero = score.count('0')
```

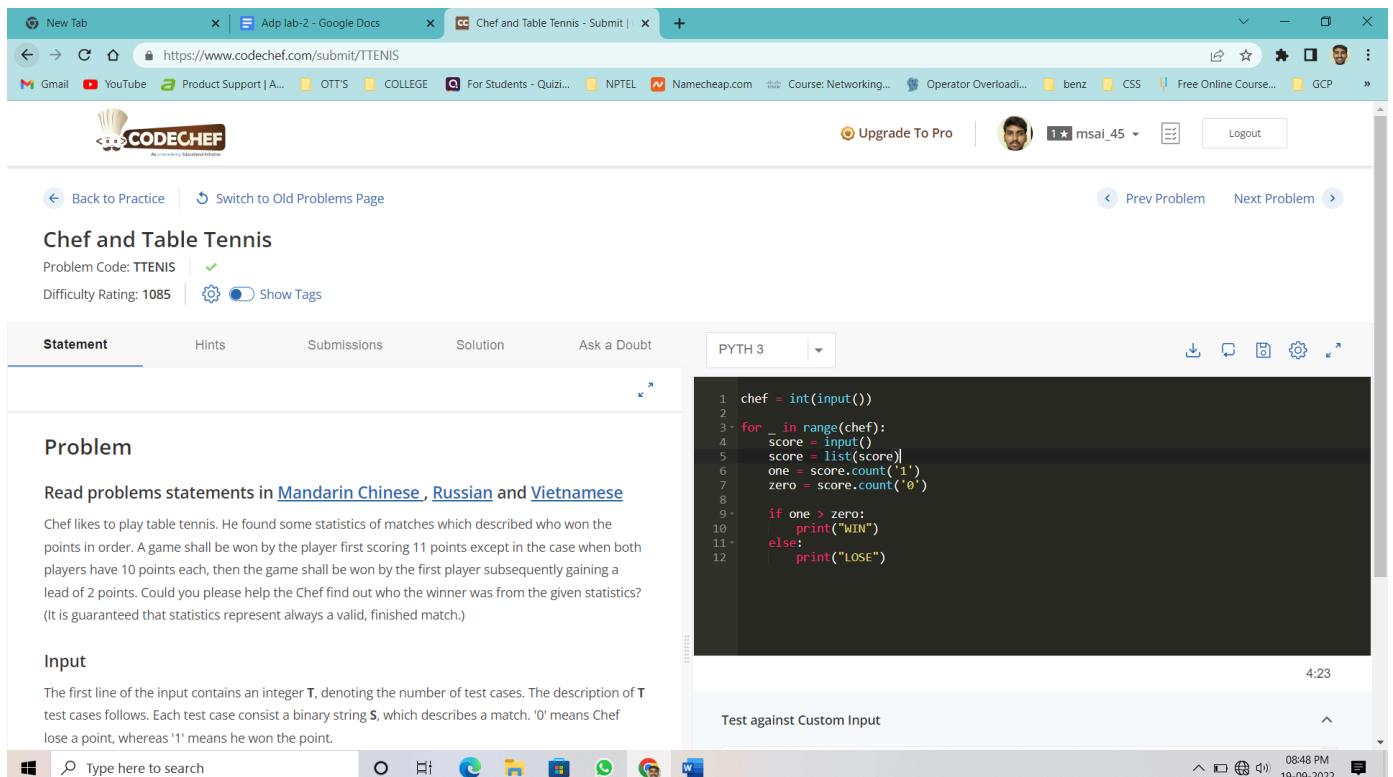
if one > zero:

```
print("WIN")
```

else:

```
print("LOSE")
```

## Output :



The screenshot shows a browser window with the following details:

- Address Bar:** https://www.codechef.com/submit/TTENIS
- Title Bar:** Chef and Table Tennis - Submit
- User Information:** msai\_45
- Code Editor (PYTH 3):**

```
1 chef = int(input())
2 for _ in range(chef):
3     score = input()
4     score = list(score)
5     one = score.count('1')
6     zero = score.count('0')
7
8     if one > zero:
9         print("WIN")
10    else:
11        print("LOSE")
```
- Bottom Status Bar:** Type here to search, Task View, File Explorer, Edge, File, Taskbar icons, 08:48 PM, 19-09-2022

The screenshot shows a browser window with the URL <https://www.codechef.com/submit/TTENIS>. The page is titled "Chef and Table Tennis - Submit". The "PYTH 3" tab is selected. The "Statement" section contains input and output descriptions. The "Input" section says: "The first line of the input contains an integer  $T$ , denoting the number of test cases. The description of  $T$  test cases follows. Each test case consists a binary string  $S$ , which describes a match. '0' means Chef lose a point, whereas '1' means he won the point." The "Output" section says: "For each test case, output on a separate line a string describing who won the match. If Chef won then print "WIN" (without quotes), otherwise print "LOSE" (without quotes)." The "Constraints" section specifies:  $1 \leq T \leq 1000$  and  $1 \leq \text{length}(S) \leq 100$ . The "Sample 1:" table shows an input of "2 010111111111 11100000000000" resulting in an output of "WIN LOSE". The "Explanation" section includes two examples: Example case 1 where Chef won 2 points and lost 1, and Example case 2 where Chef lost all 11 points. The "Test against Custom Input" section shows the input "2 010111111111 11100000000000". The "Compile & Run" and "Submit Code" buttons are visible. The status bar indicates "Status: ✓ Correct Answer", "Submission ID: 74510028", "Time: 0.03s", and "Congratulations on solving the problem. Visit our practice section to solve more interesting problems". The bottom right shows the date and time: "08:48 PM 19-09-2022".

**Time Complexity :**  $O(n)$

**Space Complexity :**  $O(n)$

**Result :** Successfully Executed the program.

## WEEK – 5

### Task – 1

**Aim :** On a sunny day, Akbar and Birbal were taking a leisurely walk in palace gardens. Suddenly, Akbar noticed a bunch of sticks on the ground and decided to test Birbal's wits.

There are  $NN$  stick holders with negligible size (numbered 11 through  $NN$ ) in a row on the ground. Akbar places all the sticks in them vertically; for each valid  $ii$ , the initial height of the stick in the  $ii$ -th holder is  $A_i$ . Birbal has a stick cutter and his task is to completely cut all these sticks, i.e. reduce the heights of all sticks to 00. He may perform zero or more operations; in each operation, he should do the following:

- Choose an integer  $HH$  and fix the cutter at the height  $HH$  above the ground.
- The cutter moves from the 11-st to the  $NN$ -th stick holder. Whenever it encounters a stick whose current height is greater than  $HH$ , it cuts this stick down to height  $HH$  (i.e. for a stick with height  $h > H$ , it removes its upper part with length  $h-H$ ).
- All the upper parts of sticks that are cut in one operation must have equal lengths. Otherwise, the operation may not be performed.

For example, if the heights of sticks are initially [5, 3, 5][5,3,5], then some valid values for  $HH$  in the first operation are 33 and 44 — the cutter cuts the upper parts of two sticks and their lengths are [2, 2][2,2] and [1, 1][1,1] respectively.  $H = 2H=2$  is an invalid choice because it would cut the upper parts of all three sticks with lengths [3, 1, 3][3,1,3], which are not all equal.

Akbar wants Birbal to completely cut all sticks in the minimum possible number of operations. If you want to be friends with Birbal, help him solve the problem.

**Sample Input :**

- The first line of the input contains a single integer  $T$  denoting the number of test cases. The description of  $T$  test cases follows.
- The first line of each test case contains a single integer  $N$ .
- The second line contains  $N$  space-separated integers  $A_1, A_2, \dots, A_N$ .

**Sample Output :**

For each test case, print a single line containing one integer — the minimum number of operations needed to completely cut all the sticks.

**Program :**

```
test = int(input())

for i in range(test):
    number = int(input())
    array = list(map(int, input().split()))

    if 0 in array:
        print(len(set(array)) - 1)
    else:
        print(len(set(array)))
```

**Time Complexity :  $O(n)$** **Space Complexity :  $O(n)$**

## Output :

**Statement**

Difficulty Rating: 1199

**Problem**

Read problem statements in [Hindi](#), [Bengali](#), [Mandarin Chinese](#), [Russian](#), and [Vietnamese](#) as well.

On a sunny day, Akbar and Birbal were taking a leisurely walk in palace gardens. Suddenly, Akbar noticed a bunch of sticks on the ground and decided to test Birbal's wits.

There are  $N$  stick holders with negligible size (numbered 1 through  $N$ ) in a row on the ground. Akbar places all the sticks in them vertically; for each valid  $i$ , the initial height of the stick in the  $i$ -th holder is  $A_i$ . Birbal has a stick cutter and his task is to completely cut all these sticks, i.e. reduce the heights of all sticks to 0. He may perform zero or more operations; in each operation, he should do the following:

- Choose an integer  $H$  and fix the cutter at the height  $H$  above the ground.
- The cutter moves from the 1-st to the  $N$ -th stick holder. Whenever it encounters a stick whose current height is greater than  $H$ , it cuts this stick down to height  $H$  (i.e. for a stick with height  $h > H$ , it removes its upper part with length  $h - H$ ).
- All the upper parts of sticks that are cut in one operation must have equal lengths. Otherwise, the operation may not be performed.

For example, if the heights of sticks are initially [5, 3, 5], then some valid values for  $H$  in the first

**Test against Custom Input**

```
1
3
1 2 3
```

**Upload code as file** **Compile & Run** **Submit Code**

9:30

**Statement**

Difficulty Rating: 1199

**Subtasks**

**Subtask #1 (20 points):**  $N \leq 50$

**Subtask #2 (80 points):** original constraints

**Sample 1:**

Input	Output
1 3 1 2 3	3

**Explanation:**

**Example case 1:** Birbal may perform the following three operations:

- Fix the cutter at  $H = 2$ . The heights of the sticks after this operation are [1, 2, 2].
- Fix the cutter at  $H = 1$ . The heights of the sticks after this operation are [1, 1, 1].
- Fix the cutter at  $H = 0$ . After this operation, all sticks are completely cut.

**Status:** ✓ Correct Answer

Submission ID: [76355797](#)

Time: 0.38s

Sub-Task	Task #	Result (time)
1	1	AC (0.023111)
Subtask Score: 20.00%		Result - AC
2	2	AC (0.383483)
Subtask Score: 80.00%		Result - AC
Total Score = 100.00%		

08:25 PM 07-10-2022

**Result : Sucessully Executed the program.**

## Task – 2

**Aim :** Chef is given a binary string AA of length NN. He can perform the following operation on AA any number of times:

- Choose LL and RR ( $1 \leq L \leq R \leq N$ ) ( $1 \leq L \leq R \leq N$ ), such that, in the substring  $A[L,R]A[L,R]$ , the number of 11s is equal to the number of 00s and reverse the substring  $A[L,R]A[L,R]$ .

Find the lexicographically smallest string that Chef can obtain after performing the above operation any (possibly zero) number of times on AA.

String XX is lexicographically smaller than string YY, if either of the following satisfies:

- XX is a prefix of YY and  $X \neq Y$ .
- There exists an index  $i$  such that  $X_i < Y_i$  and  $X_j = Y_j$ ,  $\forall j < i$ .

### Sample Input :

- First line will contain TT, the number of test cases. Then the test cases follow. Each test case contains two lines.
- The first line contains the integer NN, the length of the binary string.
- The second line contains the binary string AA.

### Sample Output :

For each test case, print the lexicographically smallest binary string that can be obtained after performing the operation any (possibly zero) number of times.

### Program :

```
test = int(input())
```

```
for _ in range(test):
```

```

length = int(input())
string_array = input()
bin_array = []
new_binary = ""

for i in range(length):
    bin_array.append(int(string_array[i]))
bin_array.sort()
for j in range(length):
    new_binary += str(bin_array[j])
print(new_binary)

```

## Output :

The screenshot shows a web-based code editor interface. At the top, there's a browser header with tabs for 'New Tab', 'Adp lab-2 - Google Docs', and 'Balanced Reversals - Submit | CodeChef'. Below the header, the page title is 'Balanced Reversals' with a difficulty rating of 'Difficulty Rating: 1165'. The main content area has tabs for 'Statement', 'Hints', 'Submissions', 'Solution', and 'Ask a Doubt'. The 'Solution' tab is active, displaying a Python code snippet:

```

1 test = int(input())
2
3 for i in range(test):
4     Length = int(input())
5     string_array = input()
6     bin_array = []
7     new_binary = ""
8
9     for i in range(Length):
10         bin_array.append(int(string_array[i]))
11     bin_array.sort()
12     for j in range(Length):
13         new_binary += str(bin_array[j])
14     print(new_binary)

```

Below the code, there's a 'Test against Custom Input' section with the following input:

```

2
5
01100
4

```

At the bottom of the editor, there are buttons for 'Upload code as file', 'Compile & Run', and 'Submit Code'. The status bar at the bottom right shows the date and time: '07-10-2022 08:48 PM'.

**Balanced Reversals** Difficulty Rating: 1165

**Statement** **Constraints**

- $1 \leq T \leq 100$
- $1 \leq N \leq 10^5$
- Sum of  $N$  over all test cases does not exceed  $2 \cdot 10^5$ .

**Sample 1:**

Input	Output
2 5 01100 4 0000	00011 0000

**Explanation:**

**Test Case 1:** Chef can choose  $L = 2$  and  $R = 5$ . The chosen substring,  $A[2, 5] = 1100$ . On reversing this, we get 0011. Thus, the final string is  $A = 00011$ . Note that this is the lexicographically smallest string possible.

**Test Case 2:** Since the string is already lexicographically minimum, Chef does not need to apply any operation.

**Status:** ✓ Correct Answer Submission ID: 76360563

Time: 0.13s

Sub-Task	Task #	Result (time)
1	1	AC (0.076463)
1	2	AC (0.131185)
1	3	AC (0.023623)

Subtask Score: 100.00% Result - AC

Total Score = 100.00%

08:50 PM 07-10-2022

## Time Complexity : $O(n^3)$

## Space Complexity : $O(n)$

## Result : Sucessfully Executed the Program.

### Task – 3

**Aim :** A Little Elephant from the Zoo of Lviv likes *lucky strings*, i.e., the strings that consist only of the lucky digits 4 and 7.

The Little Elephant has K favorite lucky strings  $A_1, A_2, \dots, A_K$ . He thinks that the lucky string S is *good* if either  $|S| \geq 47$  or for some j from 1 to K we have that  $A_j$  is a substring of S.

The Little Elephant has found N lucky strings  $B_1, B_2, \dots, B_N$  under the pillow. Now he wants to know which of them are good. Help him and find for each i from 1 to N whether the string  $B_i$  is good or not.

Let S be some lucky string. Then

- $|S|$  denotes the length of the string S;
- $S[i]$  ( $1 \leq i \leq |S|$ ) denotes the  $i^{\text{th}}$  character of S (the numeration of characters starts from 1);
- The string T of the length M is called a *substring* of S if for some k from 0 to  $|S| - M$  we have  
 $T[1] = S[k + 1], T[2] = S[k + 2], \dots, T[M] = S[k + M]$ .

#### Sample Input :

The first line of the input file contains two integers K and N, the number of favorite lucky strings of the Little Elephant and the number of strings he has found under the pillow. Each of the following K lines contains one favorite lucky string. Namely,  $j^{\text{th}}$  line among these K lines contains the string  $A_j$ . Each of the following N lines contains one lucky string that was found under the pillow. Namely,  $i^{\text{th}}$  line among these N lines contains the string  $B_i$ . The input file does not contain any whitespaces.

**Sample Output :**

For each of the N strings that were found under the pillow  
print Good if it is good, and Bad otherwise.

**Program :**

```
# cook your dish here

k, n = map(int,input().split())
a = [input() for i in range(k)]
b = [input() for i in range(n)]

for each in b:
    count = 0
    for j in a:
        if j in each:
            count = 1
    if len(each) >= 47:
        count = 1
    if count == 0:
        print('Bad')
    else:
        print('Good')
```

# Output :

### Little Elephant and Strings

Difficulty Rating: 1166    Expand

**Statement**   Submissions   Solution   Ask a Doubt

**Problem**

A Little Elephant from the Zoo of Lviv likes lucky strings, i.e., the strings that consist only of the lucky digits 4 and 7.

The Little Elephant has **K** favorite lucky strings  $A_1, A_2, \dots, A_K$ . He thinks that the lucky string  $S$  is good if either  $|S| \geq 47$  or for some  $j$  from 1 to  $K$  we have that  $A_j$  is a substring of  $S$ .

The Little Elephant has found **N** lucky strings  $B_1, B_2, \dots, B_N$  under the pillow. Now he wants to know which of them are good. Help him and find for each  $i$  from 1 to  $N$  whether the string  $B_i$  is good or not.

**Notes.**

Let  $S$  be some lucky string. Then

- $|S|$  denotes the length of the string  $S$ ;
- $S[i]$  ( $1 \leq i \leq |S|$ ) denotes the  $i^{\text{th}}$  character of  $S$  (the numeration of characters starts from 1);
- The string  $T$  of the length  $M$  is called a substring of  $S$  if for some  $k$  from 0 to  $|S| - M$  we have  $T[1] = S[k + 1], T[2] = S[k + 2], \dots, T[M] = S[k + M]$ .

**Input**

The first line of the input file contains two integers  $K$  and  $N$ , the number of favorite lucky strings of the Little Elephant and the number of strings he has found under the pillow. Each of the following  $K$

```
# cook your dish here
k, n = map(int, input().split())
a = [input() for i in range(k)]
b = [input() for i in range(n)]

for each in b:
    count = 0
    for j in a:
        if j in each:
            count = 1
    if len(each) >= 47:
        count = 1
    if count == 0:
        print('Bad')
    else:
        print('Good')
```

Test against Custom Input

```
2 4
47
744
7444
```

Upload code as file

### Little Elephant and Strings

Difficulty Rating: 1166    Expand

**Statement**   Submissions   Solution   Ask a Doubt

**Sample 1:**

Input	Output
2 4	Good
47	Good
744	Bad
7444	Good
447	
7774	
77777777777777777777777777777777	
7777777774	

**Explanation:**

The string  $S = 7444$  is good since the favorite string  $744$  is its substring.

The string  $S = 447$  is good since the favorite string  $47$  is its substring.

The string  $S = 7774$  is bad since none of the favorite strings  $47$  and  $744$  is a substring of  $S$ .

The string  $S = 77777777777777777777777777777774$  is good since its length is  $47$ . Note, however, that  $S$  does not have favorite substrings at all.

Upload code as file    

**Problem Solver Badge** 37 / 50    

Status: ✓ Correct Answer   Submission ID: 76372233

Time: 0.02s

Congratulations on solving the problem. Visit our practice section to solve more interesting problems   View another problem

**Time Complexity :  $O(n^2)$**

**Space Complexity :  $O(n)$**

**Result : Sucessfully Executed the Program.**

## WEEK – 7

### Task – 1:

**Aim :** Chef has a string  $S$  consisting of lowercase English characters. Chef defined functions left shift  $L(X)L(X)$  and right shift  $R(X)R(X)$  as follows.

- $L(X)L(X)$  is defined as shifting all characters of string  $X$  one step towards left and moving the first character to the end.
- $R(X)R(X)$  is defined as shifting all characters of string  $X$  one step towards the right and moving the last character to the beginning.

For example,  $L("abcd") = "bcda"$  and  $R("abcd") = "dabc"$

Chef wants to find out whether there exists a string  $V$  of the same length as  $S$  such that both  $L(V) = SL(V)=S$  and  $R(V) = SR(V)=S$  holds.

### Program :

```
cases = int(input())
```

```
for _ in range(cases):
```

```
    s = input()
```

```
    if len(s) > 1:
```

```
        le = s[-1] + s[:-1]
```

```
        ri = s[1:] + s[0]
```

```
        if le == ri:
```

```
            print("YES")
```

```
        else:
```

```
            print("NO")
```

else:

```
print("YES")
```

## Output :

```

cases = int(input())
for _ in range(cases):
    s = input()
    if len(s) > 1:
        le = s[-1] + s[:1]
        ri = s[1:] + s[0]
        if le == ri:
            print("YES")
        else:
            print("NO")
    else:
        print("YES")

```

Input	Output
4	YES
a	YES
ab	NO
abcd	YES
aaaaa	

**Explanation:**

- In the first test case, Chef can choose  $V = S$ .
- In the second test case, Chef can choose  $V = "ba"$  which satisfies both conditions.
- There doesn't exist any valid choice for string  $V$  in the third test case.

**Time Complexity :** O(n)

**Space Complexity :** O(n)

**Result :** Sucessfully executed the program.

### **Task – 2:**

**Aim :** Today, Chef decided to cook some delicious meals from the ingredients in his kitchen. There are  $NN$  ingredients, represented by strings  $S_1, S_2, \dots, S_N$ . Chef took all the ingredients, put them into a cauldron and mixed them up.

In the cauldron, the letters of the strings representing the ingredients completely mixed, so each letter appears in the cauldron as many times as it appeared in all the strings in total; now, any number of times, Chef can take one letter out of the cauldron (if this letter appears in the cauldron multiple times, it can be taken out that many times) and use it in a meal. A complete meal is the string "codechef". Help Chef find the maximum number of complete meals he can make!

### **Program :**

```
# cook your dish here
for i in range(int(input())):
    nputs = int(input())
    l = []
```

```
c,o,d,e,h,f = 0,0,0,0,0,0
```

```
for j in range(nputs):
```

```
    s = input()
```

```
    for item in s:
```

```
        if item == 'c':
```

```
            c+=1
```

```
        if item == 'e':
```

```
            e+=1
```

```
        if item == 'o':
```

```
            o+=1
```

```
        if item == 'd':
```

```
            d+=1
```

```
        if item == 'h':
```

```
            h+=1
```

```
        if item == 'f':
```

```
            f+=1
```

```
print(min(c//2, e//2, o, d, h, f))
```

## Outputs :

**Making a Meal** Difficulty Rating: 1214 Expand ↗

**Statement** Hints Submissions Solution Ask a Doubt

**Problem**

Read problems statements in [Hindi](#), [Mandarin Chinese](#), [Russian](#), [Vietnamese](#) and [Bengali](#) as well.

Today, Chef decided to cook some delicious meals from the ingredients in his kitchen. There are  $N$  ingredients, represented by strings  $S_1, S_2, \dots, S_N$ . Chef took all the ingredients, put them into a cauldron and mixed them up.

In the cauldron, the letters of the strings representing the ingredients completely mixed, so each letter appears in the cauldron as many times as it appeared in all the strings in total; now, any number of times, Chef can take one letter out of the cauldron (if this letter appears in the cauldron multiple times, it can be taken out that many times) and use it in a meal. A complete meal is the string "codechef". Help Chef find the maximum number of complete meals he can make!

**Input**

- The first line of the input contains a single integer  $T$  denoting the number of test cases. The description of  $T$  test cases follows.
- The first line of each test case contains a single integer  $N$ .
- $N$  lines follow. For each  $i$  ( $1 \leq i \leq N$ ), the  $i$ -th of these lines contains a single string  $S_i$ .

```
1 # cook your dish here
2 for i in range(int(input())):
3     inputs = int(input())
4     l = []
5
6     c,o,d,e,h,f = 0,0,0,0,0,0
7     for j in range(inputs):
8         s = input()
9         for item in s:
10            if item == 'c':
11                c+=1
12            if item == 'e':
13                e+=1
14            if item == 'o':
15                o+=1
16            if item == 'd':
17                d+=1
18            if item == 'h':
19                h+=1
20            if item == 'f':
21                f+=1
```

Test against Custom Input

3  
6  
codechef

Upload code as file ↩ Compile & Run Submit Code

**Making a Meal** Difficulty Rating: 1214 Expand ↗

**Statement** Hints Submissions Solution Ask a Doubt

cplusplus	5
oscar	
deck	
fee	
hat	
near	
5	
code	
hacker	
chef	
chaby	
dumbofe	
5	
codechef	
chefcode	
fehcedoc	
cceohfd	
codechef	

**PYTH 3**

```
cplusplus
oscar
```

**Problem Solver Badge** 40 / 50 Next Problem

Solve 10 more problems to get Bronze Badge

**Status:** ✓ Correct Answer Submission ID: 80879334

Time: 0.04s

Congratulations on solving the problem. Visit our practice section to solve more interesting problems View another problem ↗

Upload code as file ↩ Compile & Run Submit Code

**Time Complexity :**  $O(n^2)$

**Space Complexity :**  $O(n)$

**Result :** Sucessfully executed the program .

### **Task – 3:**

**Aim :** Chef recently solved his first problem on CodeChef. The problem he solved has  $NN$  test cases. He gets a score for his submission according to the following rules:

1. If Chef's code passes all the  $NN$  test cases, he gets 100 points.
2. If Chef's code does not pass all the test cases, but passes all the first  $M$  ( $M < N$ ) test cases, he gets  $K$  ( $K < 100$ ) points.
3. If the conditions 1 and 2 are not satisfied, Chef does not get any points (i.e his score remains at 0 points).

You are given a binary array  $A_1, A_2, \dots, A_N$  of length  $NN$ , where  $A_i = 1$  ( $i=1$ ) denotes Chef's code passed the  $i^{th}$  test case,  $A_i = 0$  ( $i=0$ ) denotes otherwise. You are also given the two integers  $M, K$ . Can you find how many points does Chef get?

### **Program :**

```
# cook your dish here
```

```
t=int(input())
```

```
for i in range(t):
```

```
n,m,k=map(int,input().split())
array = list(map(int,input().split()))
```

```
if sum(array) == n:
    print(100)
elif sum(array[:m]) == m:
    print(k)
else:
    print(0)
```

## Outputs :

The screenshot shows a browser window with multiple tabs open. The active tab is titled "A Subtask Problem - Problems". The URL is <https://www.codechef.com/submit/SUBTASK?tab=statement>. The page displays a subtask problem statement with a difficulty rating of 1217. The "Statement" tab is selected. The code editor contains the provided Python code. Below the code editor, there is a "Test against Custom Input" section with input values 4, 4 2 50, and 1 0 1 1. At the bottom right, there are "Compile & Run" and "Submit Code" buttons.

```

1 # cook your dish here
2 t=int(input())
3 for i in range(t):
4     n,m,k=map(int,input().split())
5     array = list(map(int,input().split()))
6
7     if sum(array) == n:
8         print(100)
9     elif sum(array[:m]) == m:
10        print(k)
11    else:
12        print(0)

```

The screenshot shows a browser window with three tabs: 'New Tab', 'Adp lab-2 - Google Docs', and 'A Subtask Problem - Problems'. The main content is 'A Subtask Problem' with a difficulty rating of 1217. The 'Statement' tab is selected, showing 'Sample 1' with input and output tables. The input table contains several binary strings, and the output table shows the corresponding results. To the right, the submission status is 'Status: ✓ Correct Answer' with Submission ID: 80883959. It details a single sub-task with Task # 0 (AC, 0.021948), Task # 1 (AC, 0.023151), and Task # 2 (AC, 0.023315). The subtask score is 100.00% and the total score is 100.00%. Buttons for 'Upload code as file', 'Compile & Run', and 'Submit Code' are visible at the bottom.

**Time Complexity :**  $O(n)$

**Space Complexity :**  $O(n)$

**Result :** Sucessfully executed the program.

## WEEK – 8

**Aim :** Given a valid (IPv4) IP address, return a defanged version of that IP address. A *defanged IP address* replaces every period "." with "[.]".

**Program :**

```
class Solution:
```

```
    def defangIPaddr(self, address: str) -> str:
```

```
        # Start Of Function
```

```
        addr = address
```

```
        spad = addr.split(".")
```

```
        jspad = "[.]".join(spad)
```

```
        return jspad
```

```
intest = int( input( ) )
```

```
Sol_Obj = Solution( )
```

```
Result = Sol_Obj.defangIPaddr(intest)
```

```
Print(Result)
```

## Output :

https://leetcode.com/problems/defanging-an-ip-address/submissions/

**Success** Details >

Runtime: 38 ms, faster than 81.57% of Python3 online submissions for Defanging an IP Address.

Memory Usage: 13.8 MB, less than 48.90% of Python3 online submissions for Defanging an IP Address.

Next challenges:

- [Word Squares](#)
- [Smallest String With A Given Numeric Value](#)
- [Longest Substring Of All Vowels in Order](#)

Show off your acceptance:

Time Submitted	Status	Runtime	Memory	Language
11/30/2022 17:30	Accepted	38 ms	13.8 MB	python3
11/25/2022 16:47	Accepted	35 ms	13.9 MB	python3

Problems Pick One < Prev 1108/2489 Next > Console Contribute i Run Code Submit

```

1 class Solution:
2     def defangIPaddr(self, address: str) -> str:
3         # Start Of Function
4         addr = address
5         spad = addr.split(".")
6         jspad = "[.]".join(spad)
7
8         return jspad

```

**Time Complexity :**  $O(\log n)$

**Space Complexity :**  $O(n)$

**Result :** Sucessfully executed the program .

**Aim :** A sentence is a list of words that are separated by a single space with no leading or trailing spaces. Each word consists of lowercase and uppercase English letters.

A sentence can be shuffled by appending the 1-indexed word position to each word then rearranging the words in the sentence.

- For example, the sentence "This is a sentence" can be shuffled as "sentence4 a3 is2 This1" or "is2 sentence4 This1 a3".

Given a shuffled sentence  $s$  containing no more than 9 words, reconstruct and return *the original sentence*.

### Program :

```
class Solution:
```

```
    def sortSentence(self, s: str) -> str:
```

```
        # Start of the Function
```

```
#s = "is2 sentence4 This1 a3"
```

```
ssp = s.split(" ")
```

```
# print(ssp)
```

```
ssd = {}
```

```
for item in ssp:
```

```
    if item[-1].isnumeric():
```

```
        ssd[int(item[-1])] = item[:-1]
```

```
# print(ssd)

ssd1 = sorted(ssd.keys())
res = []
for i in ssd1:
    res.append(ssd[i])

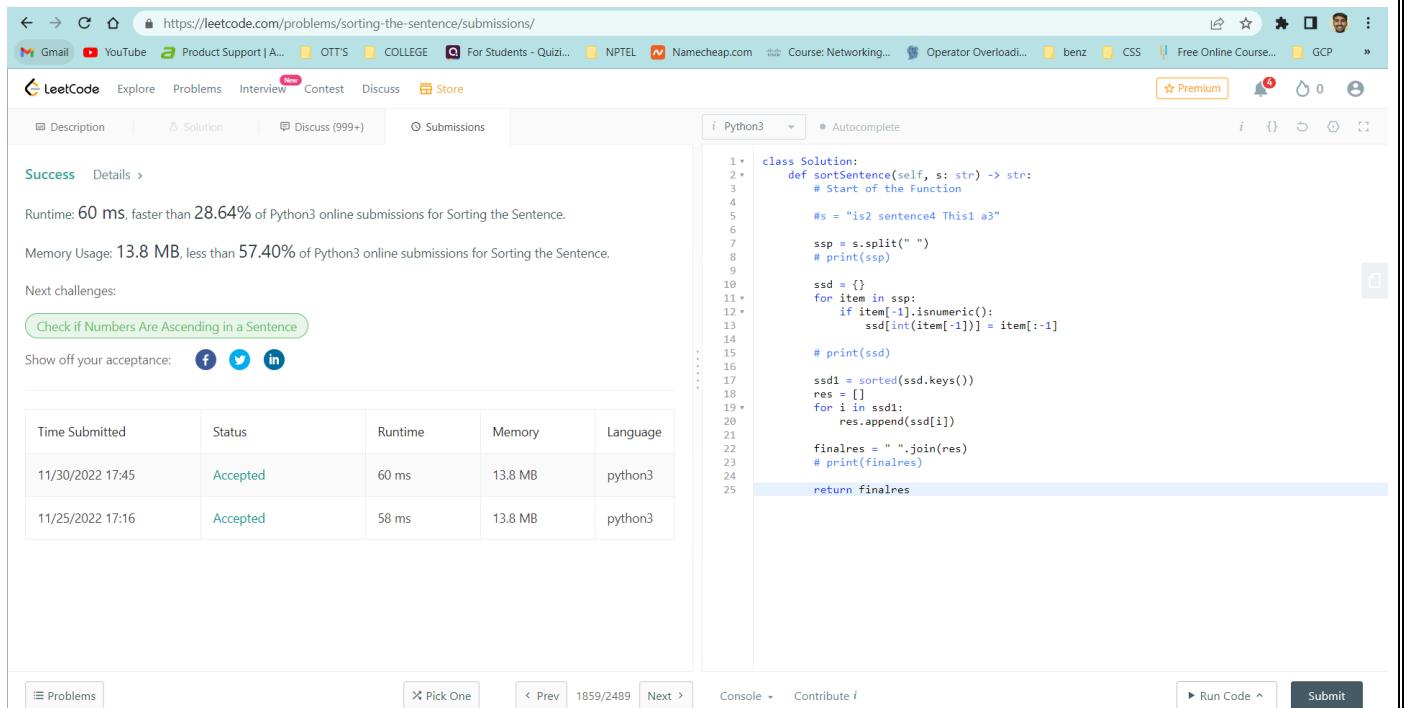
finalres = " ".join(res)
# print(finalres)

return finalres

intest = int( input( ) )
Sol_Obj = Solution( )
Result = Sol_Obj.sortSentence (intest)

Print(Result)
```

## Output :



```

1+ class Solution:
2+     def sortSentence(self, s: str) -> str:
3+         # Start of the Function
4+
5+         s = "is2 sentence4 This1 a3"
6+
7+         ssp = s.split(" ")
8+
9+         ssd = {}
10+        for item in ssp:
11+            if item[-1].isnumeric():
12+                ssd[int(item[-1])] = item[:-1]
13+
14+        # print(ssd)
15+
16+        ssd1 = sorted(ssd.keys())
17+        res = []
18+        for i in ssd1:
19+            res.append(ssd[i])
20+
21+        finalres = " ".join(res)
22+        # print(finalres)
23+
24+        return finalres
25

```

Success Details >

Runtime: 60 ms, faster than 28.64% of Python3 online submissions for Sorting the Sentence.

Memory Usage: 13.8 MB, less than 57.40% of Python3 online submissions for Sorting the Sentence.

Next challenges:

[Check if Numbers Are Ascending in a Sentence](#)

Show off your acceptance:

Time Submitted	Status	Runtime	Memory	Language
11/30/2022 17:45	Accepted	60 ms	13.8 MB	python3
11/25/2022 17:16	Accepted	58 ms	13.8 MB	python3

Problems [X Pick One](#) [◀ Prev](#) 1859/2489 [Next ▶](#) Console Contribute [i](#) [Run Code ^](#) [Submit](#)

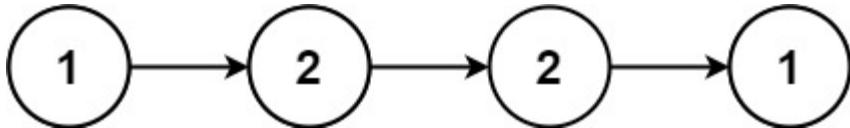
**Time Complexity : O(n)**

**Space Complexity : O(n)**

**Result : Sucessfully Executed the Program.**

## WEEK – 9

**Aim :** Given the head of a singly linked list, return true *if it is a palindrome or false otherwise.*



**Input:** head = [1,2,2,1]

**Output:** true

**Program :**

```
# Definition for singly-linked list.

# class ListNode:

#     def __init__(self, val=0, next=None):
#         self.val = val
#         self.next = next

class Solution:

    def isPalindrome(self, head: Optional[ListNode]) -> bool:
        valarr = []
        current_node = head
        while current_node is not None:
            valarr.append(current_node.val)
            current_node = current_node.next
        return valarr == valarr[::-1]
```

## Output :

The screenshot shows a LeetCode submission page for the problem "Palindrome Linked List". The code is written in Python3 and defines a class `ListNode` and a method `isPalindrome` to check if a linked list is a palindrome. The code uses a list `valarr` to store node values and compares it with its reverse.

```

1  # Definition for singly-linked list.
2  # 
3  # class ListNode:
4  #     def __init__(self, val=0, next=None):
5  #         self.val = val
6  #         self.next = next
6.+
7  class Solution:
8      def isPalindrome(self, head: Optional[ListNode]) -> bool:
9          valarr = []
10         current_node = head
11         while current_node is not None:
12             valarr.append(current_node.val)
13             current_node = current_node.next
14
15         return valarr == valarr[::-1]

```

Runtime: 1662 ms, faster than 59.46% of Python3 online submissions for Palindrome Linked List.

Memory Usage: 46.6 MB, less than 56.37% of Python3 online submissions for Palindrome Linked List.

Next challenges:

- [Palindrome Number](#)
- [Valid Palindrome](#)
- [Reverse Linked List](#)
- [Maximum Twin Sum of a Linked List](#)

Show off your acceptance:

Time Submitted	Status	Runtime	Memory	Language
11/30/2022 17:54	Accepted	1662 ms	46.6 MB	python3
11/30/2022 17:54	Accepted	869 ms	46.5 MB	python3
11/30/2022 17:51	Accepted	2208 ms	46.6 MB	python3

Testcase Run Code Result Debugger

Accepted Runtime: 59 ms

Your input: [1,2,2,1]

Output: true

Expected: true

Problems Pick One < Prev 234/2489 Next > Console Use Example Testcases Run Code Submit

**Time Complexity : O(n)**

**Space Complexity : O(n)**

**Result : Sucessfully executed the program**

**Aim :** You are given a positive integer num consisting of exactly four digits. Split num into two new integers new1 and new2 by using the digits found in num. Leading zeros are allowed in new1 and new2, and all the digits found in num must be used.

- For example, given num = 2932, you have the following digits: two 2's, one 9 and one 3. Some of the possible pairs [new1, new2] are [22, 93], [23, 92], [223, 9] and [2, 329].

Return *the minimum possible sum of new1 and new2*.

### **Example 1:**

**Input:** num = 2932

**Output:** 52

**Explanation:** Some possible pairs [new1, new2] are [29, 23], [223, 9], etc.

The minimum sum can be obtained by the pair [29, 23]:  $29 + 23 = 52$ .

### **Example 2:**

**Input:** num = 4009

**Output:** 13

**Explanation:** Some possible pairs [new1, new2] are [0, 49], [490, 0], etc.

The minimum sum can be obtained by the pair [4, 9]:  $4 + 9 = 13$ .

### **Program :**

class Solution:

```
def minimumSum(self, num: int) -> int:  
    nli = [num//1000,(num//100)%10,(num//10)%10,num%10]
```

```
#This line will convert the four digit no. into array
nli.sort() #It will sort the digits in ascending order
return (nli[0]*10+nli[3])+(nli[1]*10+nli[2])
```

#Combination of first and last and the remaining two digits will give us the minimum value

## Output :

The screenshot shows a LeetCode submission page for problem 2160. The code is pasted into the editor:

```

1 * class Solution:
2 *     def minimumSum(self, num: int) -> int:
3 *         nli = [num//1000,(num//100)%10,(num//10)%10,num%10]
4 *         #This line will convert the four digit no. into array
5 *
6 *         nli.sort() #It will sort the digits in ascending order
7 *
8 *         return (nli[0]*10+nli[3])+(nli[1]*10+nli[2])
9 *
10    #Combination of first and last and the remaining two digits will give us the minimum
value

```

Runtime: 42 ms, faster than 77.49% of Python3 online submissions for Minimum Sum of Four Digit Number After Splitting Digits.

Memory Usage: 13.9 MB, less than 57.90% of Python3 online submissions for Minimum Sum of Four Digit Number After Splitting Digits.

Next challenges:

Add Digits

Show off your acceptance: [f](#) [t](#) [in](#)

Time Submitted	Status	Runtime	Memory	Language
11/30/2022 18:08	Accepted	42 ms	13.9 MB	python3

Testcase Run Code Result Debugger

Accepted Runtime: 58 ms

Your input: 2932

Output: 52

Expected: 52

Diff

Problems Pick One < Prev 2160/2489 Next > Console Use Example Testcases Run Code Submit

**Time Complexity : O(n)**

**Space Complexity : O(n)**

**Result : Sucessfully Executed the Program.**

## WEEK – 10

**Aim :** You are given a string allowed consisting of distinct characters and an array of strings words. A string is consistent if all characters in the string appear in the string allowed.

Return *the number of consistent strings in the array words.*

**Example 1:**

**Input:** allowed = "ab", words = ["ad","bd","aaab","baa","badab"]

**Output:** 2

**Explanation:** Strings "aaab" and "baa" are consistent since they only contain characters 'a' and 'b'.

**Example 2:**

**Input:** allowed = "abc", words = ["a","b","c","ab","ac","bc","abc"]

**Output:** 7

**Explanation:** All strings are consistent.

**Program :**

class Solution:

```
def countConsistentStrings(self, allowed: str, words: List[str]) -> int:
```

```
    countie = 0
```

```
    for word in words:
```

```
        value = True
```

```
        for char in word:
```

if char not in allowed:

    value = False

if value:

    countie += 1

return countie

## Output :

The screenshot shows a LeetCode submission page for the problem "Count the Number of Consistent Strings". The code is written in Python3 and is accepted with a runtime of 66 ms. The code logic is identical to the one provided in the text above.

```
class Solution:
    def countConsistentStrings(self, allowed: str, words: List[str]) -> int:
        countie = 0
        for word in words:
            value = True
            for char in word:
                if char not in allowed:
                    value = False
            if value:
                countie += 1
        return countie
```

Testcase | Run Code | Result: Accepted | Runtime: 66 ms | Diff

Your input: "ab" ["ad", "bd", "aab", "baa", "badab"]

Output: 2

Expected: 2

Time Complexity :  $O(n)$

Space Complexity :  $O(n)$

Result : Sucessfully Executed the Program .

**Aim :** Given a positive integer n, return *the smallest positive integer that is a multiple of both 2 and n*.

**Example 1:**

**Input:** n = 5

**Output:** 10

**Explanation:** The smallest multiple of both 5 and 2 is 10.

**Example 2:**

**Input:** n = 6

**Output:** 6

**Explanation:** The smallest multiple of both 6 and 2 is 6. Note that a number is a multiple of itself.

**Program :**

```
class Solution:
```

```
    def smallestEvenMultiple(self, n: int) -> int:
```

```
        if n%2 == 0:
```

```
            return n
```

```
        return 2*n
```

## Output :

The screenshot shows a LeetCode submission page for the problem "Smallest Even Multiple". The code is written in Python3:

```
1 class Solution:
2     def smallestEvenMultiple(self, n: int) -> int:
3         if n%2 == 0:
4             return n
5         return 2*n
```

Runtime: 33 ms, faster than 89.57% of Python3 online submissions for Smallest Even Multiple.

Memory Usage: 13.9 MB, less than 6.98% of Python3 online submissions for Smallest Even Multiple.

Next challenges:

- Greatest Common Divisor of Strings
- Three Divisors
- Find Greatest Common Divisor of Array
- Convert the Temperature
- Minimum Cuts to Divide a Circle

Show off your acceptance: [f](#) [t](#) [in](#)

Time Submitted	Status	Runtime	Memory	Language
11/30/2022 18:45	Accepted	33 ms	13.9 MB	python3

Testcase Run Code Result Debugger

Accepted Runtime: 53 ms

Your input: 5

Output: 10  Diff

Expected: 10

Problems [X Pick One](#) [< Prev](#) 2413/2489 [Next >](#) Console Use Example Testcases Run Code Submit

**Time Complexity : O(n)**

**Space Complexity : O(n)**

**Result : Sucessfully executed the program.**

Date : .....

SHEET No: .....