Set-1(odd roll numbers)

## 1. [Time Machine Fun]

Gru is a super-scientist living in Gotham city in the post-apocalyptic period AD 2098. Gru has a lot of Minions in his home. He invents a time machine which will take these minions into 2020 in the Era of Covid-19. The time machine will allow Minions only based on the following conditions. Lets say N minions are having ages A= [a1, a2,…,aN]. Inside the machine, a youngest minion can start handshaking with others at a time. After the handshake, their ages will be decremented by the age of the youngest minion. In this way the age of minions can be updated. If all of their ages become 0s at the end of a fixed number of handshakes, then the time machine will take them to 2020 in the Era of Covid-19. Otherwise the time machine will give each of the minions a punch in their faces.

Let's say A = [1, 2, 1, 2, 2], after the 1st handshake operation their ages are decremented by one and then the array becomes [0, 1, 0, 1, 1]. If 2nd handshake then A=[0, 0, 0, 0, 0]. So the answer should be "Yes" for N=2 and the answer should be "No" for N=1.
If A=[1, 2, 3, 4, 5] and N=4 then the answer should be "No".
If A=[1, 2, 3, 4, 5] and N=5 then the answer should be "Yes".

Given the ages of minions A you need to tell Gru whether they can make it to 2020 or not.

**Sample Test Cases:**
**Input:**
5
1 2 3 4 5
8
**Output:**
Yes

**Input:**
5
1 2 3 4 5
4
**Output:**
No

**Input:**
4
1 1 2 3
3
**Output:**
Yes

where,

- First line represents the number of minions/elements.
- Second line represents the ages of all minions.
- Third line represents number of handshakes.

2. **[Gaming Fun]**

In a free class, Ms. Rita decided to have some fun activity with the students. She made a block of R x C and some chits having written ‘z’, ‘zig’ and random values. If the student picks up the chit having ‘z’ then first the block is printed and then he has to traverse the R X C block in z manner, if the chit has ‘zig’ written on it then then first the block is printed and student has to move in zigzag manner and for any other chit, print the block and then 'invalid choice' is printed.

*Testcase1:*

**Input:** (row, column,choice, values for rxc)

3
3
z
1
2
3
4
5
6
7
8
9

**Output:**

1 2 3
4 5 6
7 8 9
Z traversal
1 2 3 5 7 8 9

*Testcase2:*

**Input:** (row, column,choice, values for rxc)

3
4
zig
10
20
30
40
50
60
70

80
90
100
110
120
**Output:**
10 20 30 40
50 60 70 80
90 100 110 120
Zigzag Traversal
10 20 30 40 80 70 60 50 90 100 110 120

*Testcase3:*
**Input:** (row, column,choice, values for rxc)
3
3
alacarte
1
2
3
4
5
6
7
8
9
**Output:**
1 2 3
4 5 6
7 8 9
Invalid choice

where,

- First line represents the number elements in row R
- Second line represents the number elements in column C
- Third line represents the choice
- Followed by elements R x C

**3. [I-Spy game]**

In a multiplayer spy game, players can act either as a spy or a thief. Assume there are N number of players standing in a row. A spy can catch a thief as per the following conditions:
1. Each spy can catch only one thief.

2. A spy has a search range K, which means he can not catch a thief who is more than K positions away from him in the row.

For a given game scenario you are supposed to tell how many thieves can be caught. The total number of players, spy search range and player arrangement are given as input for you.

***Test Case 1***

**Input:**

5 # total number of players
1 # search range of a spy
S T T S T # player arrangement in a row (space separated)

**Output:**

2 # Here search range is 1,thus first spy (position 0) catches the first thief (at position 1) and second spy(at position 3) can catch either the second or third thief (at position 2 and 4).

***Test Case 2***

**Input:**

4
2
S T T S

**Output:**

2 # now the search range is 2 thus both thieves can be caught.

**4. [Another Game of token]**

You are playing a computer game where a random bit generator box produces a bit ( 0 or 1) every time one presses the button. You selected a preferred bit at the beginning and played several rounds. In a round, if your preferred bit is produced by the generator box then you win else lose. The outcome string of 0 and 1 represented the bit generated in each round.
You will start the game with a total cash of 100 Rs. Each round has a token amount of Rs 10. The detailed rules for the game is as follows:

**a)** If you win a round, the token will be added to your current sum and the token amount for next round will become 10.
**b)** If you lose a round, the token amount will be reduced from your current sum and the next round token amount will become twice the previous.
**c)** The game ends either when all the rounds are complete or when you don't have sufficient sum.
You need to find and print the amount at the end of the game (final sum) and in case you do not have enough money in between the game to play the next round, then print -1.
**Note: For simplicity let's assume that you always choose 1 as your preferred bit for win.**

***Testcase1***
**Input:**
4 # No of rounds
1110 # outcome of each round( number of rounds can vary)
**Output:**
120

**Explanations:** 3 win and one loss, thus 100+10+10+10-10=120

***Testcase2***
**Input:**
3# No of rounds
100
**Output:**
80 # 1 win and 2 loss, thus 100+10-10-20 ( token price doubled after a loss) = 80

***Testcase3***
**Input:**
4# No of rounds
0000
**Output:**
-1 # 4 loss, thus 100 -10 -20 -40 -80 ( token price doubles for each loss) is < 0 thus -1 output

***Testcase4***
**Input:**
4# No of rounds
0010
**Output:**
100 # 100 -10-20+40-10 (after each loss, token amount doubles and after each win reset to 10)