Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Monday, 25 November 2024, 7:11 PM
Duration	27 days 22 hours

Question **1**

Correct

Marked out of 3.00

⚑ Flag question

A set of N numbers (separated by one space) is passed as input to the program. The program must identify the count of numbers where the number is odd number.

Input Format:

The first line will contain the N numbers separated by one space.

Boundary Conditions:

3 <= N <= 50

The value of the numbers can be from -99999999 to 99999999

Output Format:

The count of numbers where the numbers are odd numbers.

Example Input / Output 1:

Input:

5 10 15 20 25 30 35 40 45 50

Output:

5

Explanation:

The numbers meeting the criteria are 5, 15, 25, 35, 45.

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int
    num,count=0,num_count=0;
    while((scanf("%d",&num)==1))
    {
        if(num%2!=0)
        {
            count++;
        }
        else{
            num_count++;
        }
    }
    printf("%d",count);
    return 0;
}
```

	Input	Expected	Got	
✓	5 10 15 20 25 30 35 40 45 50	5	5	✓

Passed all tests! ✓

Given a number N, return true if and only if it is a *confusing number*, which satisfies the following condition:

We can rotate digits by 180 degrees to form new digits. When 0, 1, 6, 8, 9 are rotated 180 degrees, they become 0, 1, 9, 8, 6 respectively. When 2, 3, 4, 5 and 7 are rotated 180 degrees, they become invalid. A *confusing number* is a number that when rotated 180 degrees becomes a **different** number with each digit valid.

Example 1:

6 -> 9

Input: 6

Output: true

Explanation:

We get 9 after rotating 6, 9 is a valid number and 9!=6.

Example 2:

89 -> 68

Input: 89

Output: true

Explanation:

We get 68 after rotating 89, 86 is a valid number and 86!=89.

Note:

1. 0 <= N <= 10^9

2. After the rotation we can ignore leading zeros, for example if after rotation we have 0008 then this number is considered as just 8.

Answer: (penalty regime: 0 %)

```c
1  #include<stdio.h>
2  int main()
3  {
4      int a,ch;
5      scanf("%d",&a);
6      while(a!=0){
7          int b=a%10;
8          a=a/10;
9          switch(b){
10             case 0:
11             case 6:
12             case 8:
13             case 9:
14                 ch=0;
15                 break;
16             default:
17                 ch=1;
18         }
19     }
20     if(ch==1){
21         printf("false");
22     }
23     else{
24         printf("true");
25     }
26     return 0;
27 }
```

```
26        return 0;
27  }
```

	Input	Expected	Got	
✓	6	true	true	✓
✓	89	true	true	✓
✓	25	false	false	✓

Passed all tests! ✓

A nutritionist is labeling all the best power foods in the market. Every food item arranged in a single line, will have a value beginning from 1 and increasing by 1 for each, until all items have a value associated with them. An item's value is the same as the number of macronutrients it has. For example, food item with value 1 has 1 macronutrient, food item with value 2 has 2 macronutrients, and incrementing in this fashion.

The nutritionist has to recommend the best combination to patients, i.e. maximum total of macronutrients. However, the nutritionist must avoid prescribing a particular sum of macronutrients (an 'unhealthy' number), and this sum is known. The nutritionist chooses food items in the increasing order of their value. Compute the highest total of macronutrients that can be prescribed to a patient, without the sum matching the given 'unhealthy' number.

Here's an illustration:

Given *4* food items (hence value: *1,2,3* and *4*), and the unhealthy sum being *6* macronutrients, on choosing items *1, 2, 3* -> the sum is *6*, which matches the 'unhealthy' sum. Hence, one of the three needs to be skipped. Thus, the best combination is from among:

·   *2 + 3 + 4 = 9*

·   *1 + 3 + 4 = 8*

·   *1 + 2 + 4 = 7*

Since *2 + 3 + 4 = 9*, allows for maximum number of macronutrients, 9 is the right answer.

Complete the code in the editor below. It must return an integer that represents the maximum total of macronutrients, modulo *1000000007* $(10^9 + 7)$.

Constraints

- $1 \le n \le 2 \times 10^9$
- $1 \le k \le 4 \times 10^{15}$

Input Format For Custom Testing

The first line contains an integer, n, that denotes the number of food items.

The second line contains an integer, k, that denotes the unhealthy number.

Sample Input 0

2
2

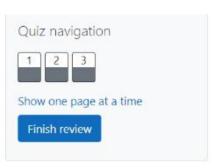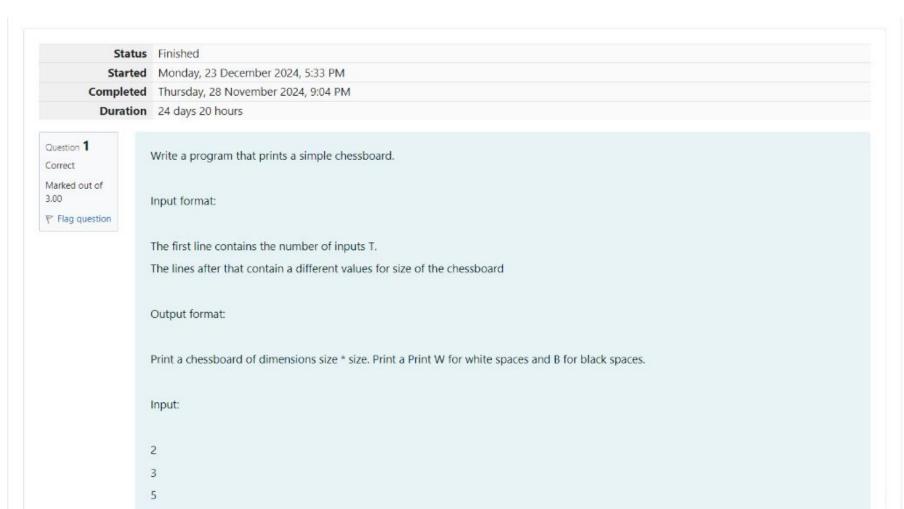**Sample Output 0**

3

```c
#include<stdio.h>
int main(){
    long int n,k,sum;
    scanf("%ld %ld",&n,&k);
    sum=0;
    for(int i=1;i<=n;i++){
        sum=sum+i;
        if(sum==k){
            sum=sum-1;
        }
    }
    printf("%ld",sum%1000000007);
}
```

	Input	Expected	Got	
✓	2 2	3	3	✓
✓	2 1	2	2	✓
✓	3 3	5	5	✓

Finish review

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Thursday, 28 November 2024, 9:04 PM
Duration	24 days 20 hours

Question **1**

Correct

Marked out of 3.00

⚑ Flag question

Write a program that prints a simple chessboard.

Input format:

The first line contains the number of inputs T.

The lines after that contain a different values for size of the chessboard

Output format:

Print a chessboard of dimensions size * size. Print a Print W for white spaces and B for black spaces.

Input:

2

3

5

Input:

2

3

5

Output:

WBW

BWB

WBW

WBWBW

BWBWB

WBWBW

BWBWB

WBWBW

Answer: (penalty regime: 0 %)

Answer: (penalty regime: 0 %)

```c
1   #include<stdio.h>
2   int main(){
3       int T,d,i=0,i1,i2;
4       char c;
5       scanf("%d",&T);
6       while(i<T)
7       {
8           scanf("%d",&d);
9           i1=0;
10          while(i1<d)
11          {
12              i2=0;
13              while(i2<d)
14              {
15                  c='B';
16                  if((i1+i2)%2==0)
17                  {
18                      c='W';
19                  }
20                  printf("%c",c);
21                  i2++;
22              }
23              i1++;
24              printf("\n");
25          }
26          i++;
27      }
28      return 0;
29  }
```

	Input	Expected	Got	
✓	2 3 5	WBW BWB WBW WBWBW BWBWB WBWBW BWBWB WBWBW	WBW BWB WBW WBWBW BWBWB WBWBW BWBWB WBWBW	✓

Passed all tests! ✓

Let's print a chessboard!

Write a program that takes input:

The first line contains T, the number of test cases

Each test case contains an integer N and also the starting character of the chessboard

Output Format

Print the chessboard as per the given examples

Sample Input / Output

Input:

2

2 W

3 B

Output:

WB

BW

BWB

WBW

BWB

Answer: (penalty regime: 0 %)

```c
1   #include<stdio.h>
2   int main()
3   {
4       int t,d,i,i1,i2,o,z;
5       char c,s;
6       scanf("%d",&t);
7       for(i=0;i<t;i++)
8       {
9           scanf("%d %c",&d,&s);
10          for(i1=0;i1<d;i1++)
11          {
12              z=(s=='W') ? 0:1;
13              o=(i1%2==z) ? 0:1;
14              for(i2=0;i2<d;i2++)
15              {
16                  c=(i2%2==o)?'W':'B';
17                  printf("%c",c);
18              }
19              printf("\n");
20          }
21      }
22      return 0;
23  }
```

	Input	Expected	Got	
✓	2 2 W 3 B	WB BW BWB WBW BWB	WB BW BWB WBW BWB	✓

Passed all tests! ✓

Decode the logic and print the Pattern that corresponds to given input.

If N= 3

then pattern will be :

10203010011012

**4050809

****607

If N= 4, then pattern will be:

1020304017018019020

**50607014015016

****809012013

******10011

Constraints

2 <= N <= 100

Input Format

First line contains T, the number of test cases

Each test case contains a single integer N

Output

First line print Case #i where i is the test case number

In the subsequent line, print the pattern

Test Case 1

3
3
4
5

Output

Case #1
10203010011012
**4050809

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int v,c=0;
    scanf("%d",&v);
    while(v!=0){
        c++;
        int a;
        scanf("%d",&a);
        int s1=10,s2=(a*(a*10))+10;
        printf("Case #%d\n",c);
        for (int i=0;i<a;i++){
            for(int j=0;j<i;j++){
                printf("**");
            }
            for(int j=0;j<a-i;j++){
                printf("%d",s1);
                s1+=10;
            }
            for(int j=0;j<a-i;j++){
                if((j+1)==(a-i)){
                    printf("%d",((s2+(j*10))/10));
                }
                else{
                    printf("%d",(s2+(j*10)));
                }
            }
            s2 -=(a-i)*10;
            s2+=10;
            printf("\n");
        }
        v--;
    }
}
```

```
31        }
32        v--;
33    }
34 }
```

	Input	Expected	Got	
✓	3 3 4 5	Case #1 10203010011012 **4050809 ****607 Case #2 1020304017018019020 **50607014015016 ****809012013 ******10011 Case #3 10203040502602702802903 0 **6070809022023024025 ****100110120190200 21 ******13014017018 ********15016	Case #1 10203010011012 **4050809 ****607 Case #2 1020304017018019020 **50607014015016 ****809012013 ******10011 Case #3 10203040502602702802903 0 **6070809022023024025 ****100110120190200 21 ******13014017018 ********15016	✓

Passed all tests! ✓