SRISHA S

B.E COMPUTER SCIENCE & DESIGN

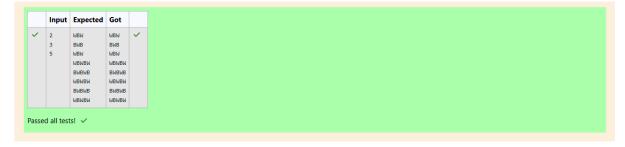
241701056

WEEK 5 **Problem Statement 1:** 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 **Output: WBW BWB WBW WBWBW BWBWB**

WBWBW

BWBWB

WBWBW



Problem Statement 2:

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:

2

2 W

3 B

Sample Output:

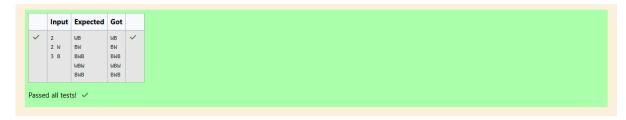
WB

BW

BWB

WBW

BWB



Problem Statement 3:

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

If N= 3 t

hen 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 Format
First line print Case #i where i is the test case number, In the
subsequent line, print the pattern
Sample Input
3
3
4
5
Sample Output
Case #1
10203010011012
**4050809
****607
Case #2
1020304017018019020
**50607014015016
****809012013
*****10011
```

Case #3

102030405026027028029030

**6070809022023024025

****10011012019020021

*****13014017018

******15016

```
Answer: (penalty regime: 0 %)
   1 #include<stdio.h>
              int t;
scanf("%d",&t);
for(int k=1;k<=t;k++)</pre>
                    int n;
scanf("%d",&n);
printf("Case #%d\n",k);
int f=1,b=n*(n+1);
    11
   12
13 ·
14
15 ·
                     for (int i=0;i<n;i++)
                    {
    for (int j=0; j<2*i; j++)
                               printf("*");
                       printf("%d",f);
f++;
for (int x=2;x<=(n-i);x++)
{</pre>
   18
   19
                           printf("0%d",f);
f++;
   22
23
24
25
                    for(int 1-0 )
{
    printf("0%d",1);
}
--i:
                          for(int l=b-(n-i)+1;1<=b;1++)
   26 v
27
28
29
                  b-=n-i;
printf("\n");
}
   30
31
32
   33 }
          Input Expected
                   Case #1
                                                     Case #1
                   10203010011012
**4050809
                                                     10203010011012
**4050809
                   ****607
                                                    ****607
                   Case #2
                                                     Case #2
                  1020304017018019020
**50607014015016
                                                    1020304017018019020
**50607014015016
                   ****809012013
                                                    ****809012013
                                                     *****10011
                   *****10011
                   Case #3
                                                     Case #3
                   102030405026027028029030 102030405026027028029030
                  *******15016

***6778899222623924025

***6778899022623924025

****10011012019020021

****13014017018

******15016
```

Problem Statement 4:

The k-digit number N is an Armstrong number if and only if the k-th power of each digit sums to N.

Given a positive integer N, return true if and only if it is an Armstrong number.

Passed all tests! ✓

Sample Input:
153
Sample Output:
True
Explanation:
153 is a 3-digit number, and 153 = 1^3 + 5^3 + 3^3.
Sample Input:
123
Sample Output:
false
Explanation:
123 is a 3-digit number, and 123 != 1^3 + 2^3 + 3^3 = 36
Sample Input:
1634
Sample Output:
True

Note:

1 <= N <= 10^8



Problem Statement 5:

Take a number, reverse it and add it to the original number until the obtained number is a palindrome.

Constraints

Passed all tests! 🗸

1<=num<=999999999

Sample Input 1:

32

Sample Output:

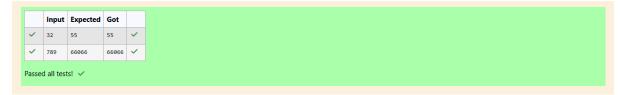
55

Sample Input 2:

789

Sample Output 2:

66066



Problem Statement 6:

A number is considered lucky if it contains either 3 or 4 or 3 and 4 both in it. Write a program to print the nth lucky number. Example, 1st lucky number is 3, and 2nd lucky number is 4 and 3rd lucky number is 33 and 4th lucky number is 34 and so on. Note that 13, 40 etc., are not lucky as they have other numbers in it.

The program should accept a number 'n' as input and display the nth lucky number as output.

Sample Input 1:

3

Sample Output 1:

33

Explanation:

Here the lucky numbers are 3, 4, 33, 34., and the 3rd lucky number is 33.

Sample Input 2:

34

Sample Output 2:

33344