Problem Statement:
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 value for size of the chessboard
Output format:
Print a chessboard of dimensions size * size.
Print W for white spaces and B for black spaces.
Sample Input:
2
3
5
Sample Output:
WBW
BWB
WBW
WBWBW
BWBWB
WBWBW
BWBWB
WBWBW

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
int main(){
    int t,n;
    scanf("%d",&t);
    while(t--){
        scanf("%d",&n);
        for(int i=0;i<n;i++){
            for(int j=0;j<n;j++){
                if((i+j)%2==0){
                    printf("W");
                }
                else{
                    printf("B");
                }
            }
            printf("\n");
        }
    }
}
```

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

Passed all tests! ✓

Problem Statement:

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:

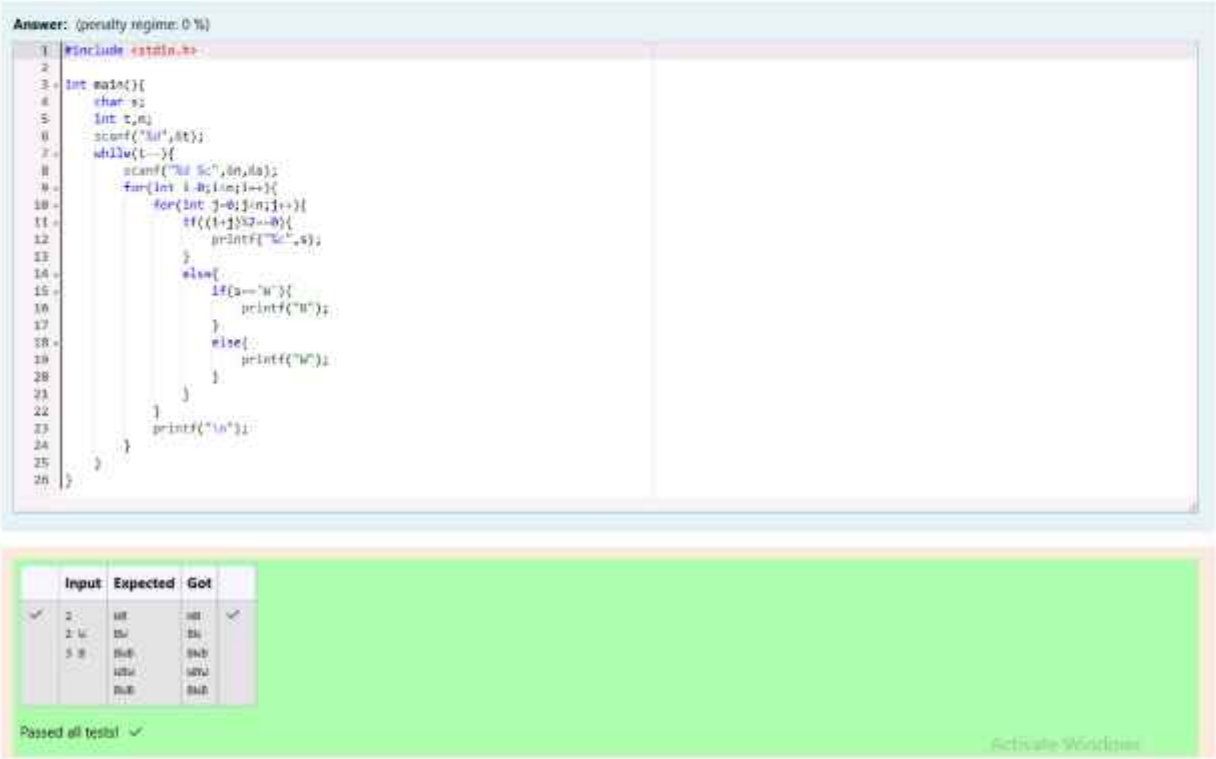
WB

BW

BWB

WBW

BWB



Problem Statement:
Decode the logic and print the Pattern that corresponds to given input.
If N= 3 then pattern will be:
102030100110 12
**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

10203040170180190 20

**506070140150 16

****809012013

******10011

Case #3

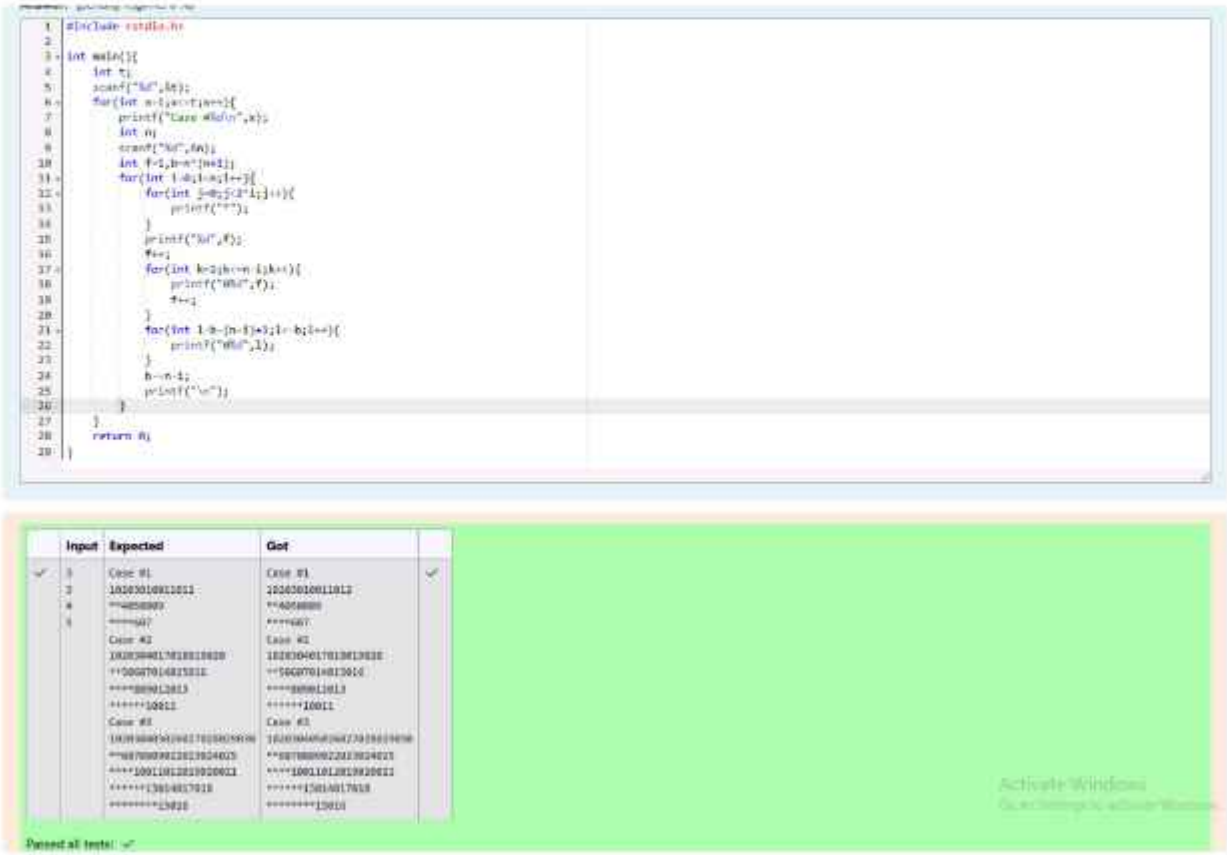
1020304050260270280290 30

**60708090220230240 25

****100110120190200 21

******13014017018

********15016



Problem Statement:

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.

Note: 1 <= N <= 10^8

Hint: 153 is a 3-digit number, and 153 = 1^3 + 5^3 + 3^3.

Sample Input:

153

Sample Output:

true

Sample Input:

123

Sample Output:

false
Sample Input:
1634
Sample Output:
true

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
#include <math.h>

int main(){
    int n,sum=0,t,c=0;
    scanf("%d",&n);
    t=n;
    while(n>0){
        n=n/10;
        c=c+1;
    }
    n=t;
    while(t>0){
        sum=sum+pow((t%10),c);
        t/=10;
    }
    if(sum==n){
        printf("true");
    }
    else{
        printf("false");
    }
}
```

	Input	Expected	Got	
✓	153	true	true	✓
✓	123	false	false	✓

Passed all tests! ✓

Problem Statement:
Take a number, reverse it and add it to the original number until the obtained number is
a palindrome.
Constraints
1<=num<=99999999
Sample Input 1
32
Sample Output 1
55

Sample Input 2
789
Sample Output 2
66066

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>


int rev(int n){
    int reverse=0;
    while(n>0){
    reverse=(reverse*10)+(n%10);
    n=n/10;
    }
    return reverse;
}
int Pal(int n){
    return rev(n)==n;
}
int main(){
    int n;
    scanf("%d",&n);
    while(!Pal(n)){
        int r=rev(n);
        n=n+r;
    }
    printf("%d",n);
}
```

Problem Statement:
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

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int main(){
    int n=1;
    int i=0,ns,c=0,e;
    scanf("%d",&e);
    while(i<e){
        ns=n;
        while(ns!=0){
            c=0;
            if(ns%10!=3&&ns%10!=4){
                c=1;
                break;
            }
            ns/=10;
        }
        if(c==0){
            i++;
        }
        n++;
    }
    printf("%d",--n);
}
```

	Input	Expected	Got	
✓	34	33344	33344	✓

Passed all tests! ✓