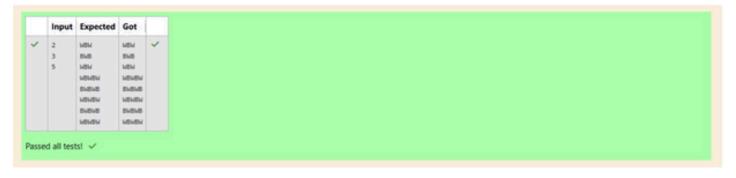
MOWNISWARAN.R
ECE-D
240801209
Problem Statement 1:
Write a program that prints a simple chessboard.
Input format:
The frst line contains the number of inputs T.
The lines after that contain a different value for size of the chessboard
Output format:
Print a chessboard of dimensions size * size. Print W for white spaces and B for black spaces. Sample Input:

3	
5	
Sample Output:	
WBW	
BWB	
WBW	
WBWBW	
BWBWB	

WBWBW

BWBWB

WBWBW

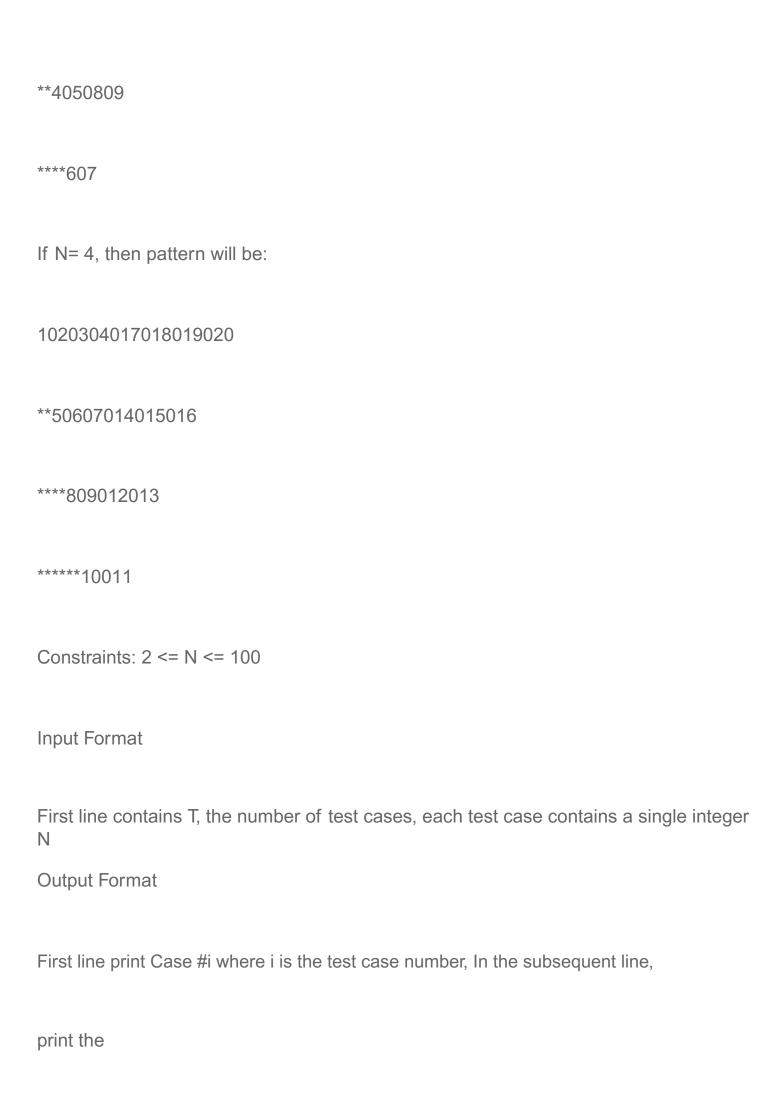


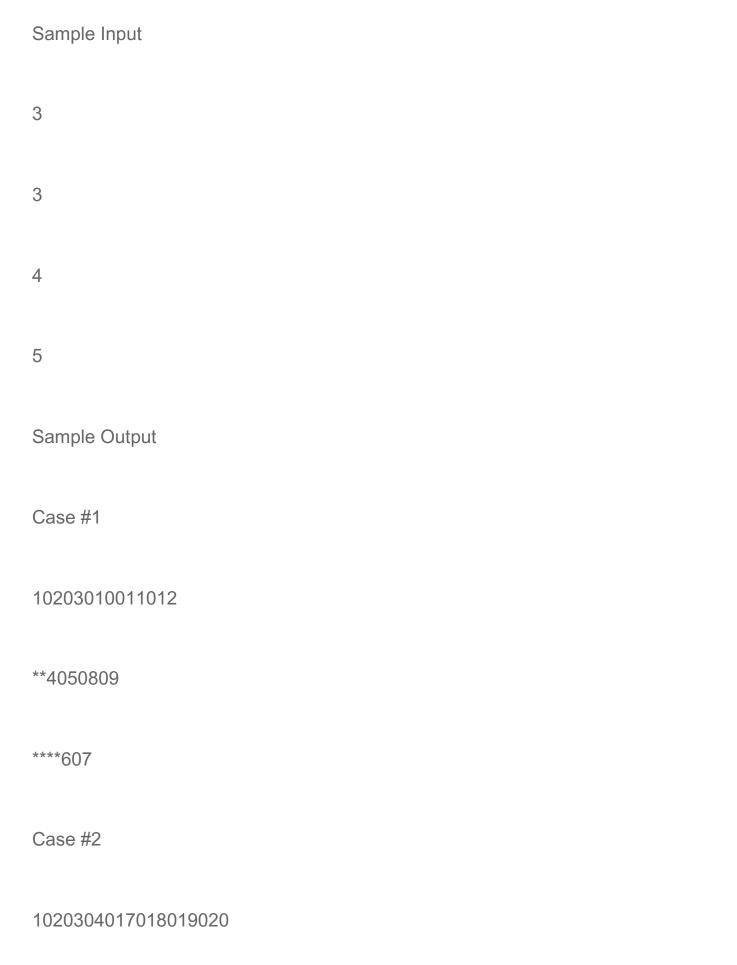
Problem Statement 2:
Let's print a chessboard!
Write a program that takes input:
The frst line contains T, the number of test cases
Each test case contains an integer N and also the starting character of the
chessboard

Output Format
Print the chessboard as per the given examples
Sample Input:
2
• W
3 B
Sample Output:
WB BW BWB WBW BWB

	Expected	Got	
/ 2	MS	ыв	,
2 W	BM	BM	
3 8	848	BMB	
	MBM	MBN	
	8145	8148	

Problem Statement 3:
Decode the logic and print the Pattern that corresponds to given input.
If N= 3 then pattern will be:
10203010011012





pattern



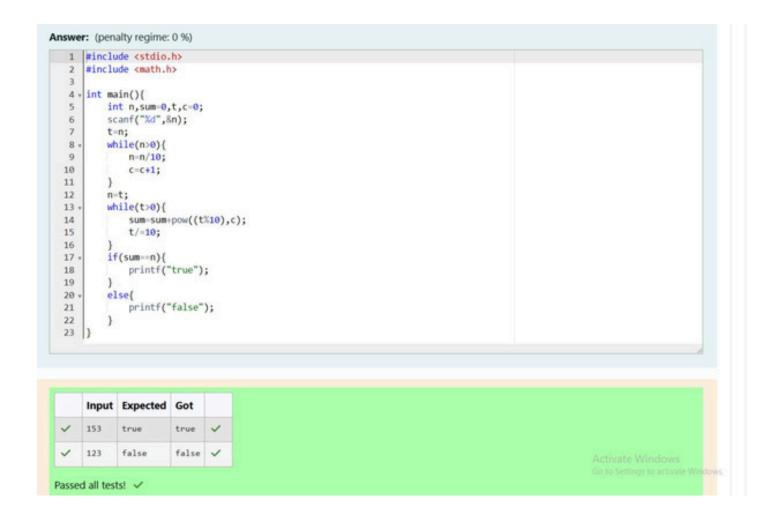


Problem Statement 4:
The k-digit number N is an Armstrong number if and only if the k-th power of
each digit

sums to N.
Given a positive integer N, return true if and only if it is an Armstrong number. Note: $1 \le N \le 10^8$
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:

Sample Output:

true



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

• 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 %)
 1 #include <stdio.h>
   3
   4 - int rev(int n){
   5
         int reverse=0;
   6 +
          while(n>0){
         reverse=(reverse*10)+(n%10);
   7
   8
         n=n/10;
   9
  10
          return reverse;
  11 }
  12 - int Pal(int n){
          return rev(n) -- n;
  13
  14 }
15 • int main(){
         int n;
scanf("%d",&n);
  16
  17
  18 +
          while(!Pal(n)){
              int r=rev(n);
  19
  20
             n=n+r;
  21
          printf("%d",n);
  22
  23 }
```

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 %)
   1 #include <stdio.h>
   3 - int main(){
           int n=1;
   4
           int i=0,ns,c=0,e;
scanf("%d",&e);
while(i<e){
   5
   6
   7 .
               ns=n;
while(ns!=0){
   8
   9 .
   10
                     C=0;
                     if(ns%10|=3&&ns%10|=4){
   11 +
                        c=1;
break;
   12
  13
   14
                     ns/=10;
   15
  16
                if(c==0){
   17 ,
                    i++;
  18
  19
   20
                n++;
   21
            printf("%d", --n);
   22
   23 }
```

