Nicholas Samuel

Question 1:	Simple Chessboard
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 %)
  9
10
11
12
  13
14
15
16
17
18
19
20 }
                }
                printf("\n");
      Input Expected Got
           WBW
                     WBW
           BWB
                     BWB
           WBW
           WBWBW
                    WBWBW
           BWBWB
                    BWBWB
           WBWBW
                    WBWBW
            BWBWB
                     BWBWB
           WBWBW
                    WBWBW
Passed all tests! 🗸
```

Question 2: Print Our Own Chessboard

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

Question 3:

Pattern Printing

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 Format** First line print Case #i where i is the test case number, In the subsequent line, print the pattern Sample Input 3345

Sample Output

Case #1

10203010011012

**4050809

****607

Case #2

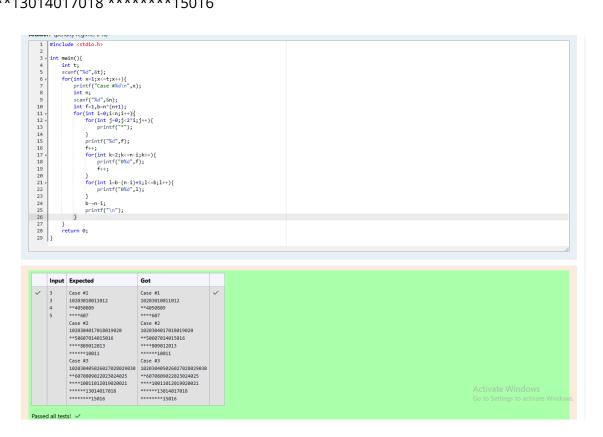
1020304017018019020

**50607014015016

****809012013

*****10011

Case #3 102030405026027028029030 **6070809022023024025 ****10011012019020021 ******13014017018 *******15016

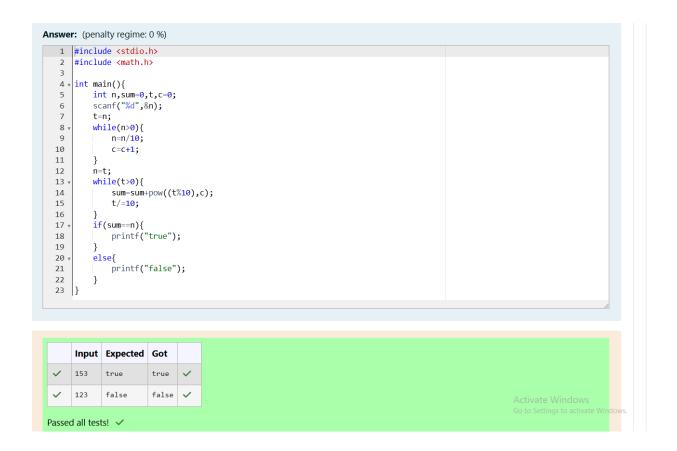


Question 4:

Armstrong Number

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



Question 5: Reverse and Add Until Get a Palindrome

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

Sample Input 2

789

Sample Output 2

66066

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

Question 6:

Lucky Number

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:

Sample Output 1:

```
Answer: (penalty regime: 0 %)
     1 #include <stdio.h>
           int main(){
    int n=1;
    int i=0,ns,c=0,e;
    scanf("%d",&e);
    while(i<e){</pre>
     3 *
     4
5
      6
                      le(i<e,,
    ns=n;
    while(ns!=0){
        c=0;
        if(ns%10!=3&&ns%10!=4){
            c=1;
            c=1;
            c=1;</pre>
      8
     9 ,
    10
    11 1
    12
    13
    14
    15
                              ns/=10;
                        }
if(c==0){
    16
    17
    18
                              i++;
    19
    20
                        n++;
    21
    22
                  printf("%d",--n);
    23 }
           Input Expected Got
                      33344
                                      33344 🗸
 Passed all tests! 🗸
```