HARSHAVARDHINI.M

CSD

241701018

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

SampleOutput:

WBW

BWB

WBW

WBWBW

BWBWB

WBWBW

BWBWB

WBWBW



WBWBW

Passed all tests! ✓

WBWBW

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

2W

3 *B*

SampleOutput:

WB

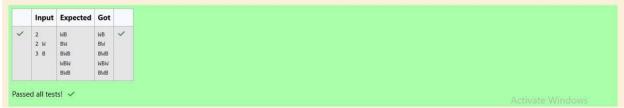
BW

BWB

WBW

BWB





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

10203010011012

**4050809

****607

IfN=4,thenpatternwillbe: 1020304017018019020

**50607014015016

****809012013

*****10011

Constraints:2<=N<=100 Input

Format

Firstline contains T, the number of test cases, each test case contains a single integer N Output Format

Firstline printCase #i where i isthe testcase number,In the subsequentline, print the pattern Sample

```
Input 3
3
4
5
SampleOutput
Case #1
10203010011012
**4050809
****607
Case #2
1020304017018019020
**50607014015016
****809012013
*****10011
Case #3
102030405026027028029030
**6070809022023024025
****10011012019020021
*****13014017018
******15016
```

```
#include <stdio.h>
          : main(){
   int t;
   scanf("%d",&t);
   for(int x=1;x<=t;x++){
      printf("case #%d\n",x);
      int n;
      scanf("%d",&n);
      int f=1;b=n*(n+1);
      for(int i=0;ion;i++){
        for(int i=0;ion;i++){
            printf("a");
      }
      **"%d",f);</pre>
         int main(){
12 * 13 14 15 16 17 * 18 19 20 21 * 22 23 24 25 26 27 28 29 ]
                                  f++;
for(int k=2;k<=n-i;k++){
    printf("0%d",f);
    f++;</pre>
                                   }
for(int l=b-(n-i)+1;l<=b;l++){
    printf("0%d",l);
}</pre>
         Input Expected
                       Case #1
10203010011012
                                                                          Case #1
10203010011012
                        **4050809
                                                                           **4050809
                                                                          ****607
                       Case #2
                                                                           Case #2
                       1020304017018019020
**50607014015016
                                                                           1020304017018019020
**50607014015016
                       ****809012013
*****10011
                                                                         ****809012013
*****10011
                       Case #3
                                                                          Case #3
                       102030405026027028029030 102030405026027028029030
                                                                        **6070809022023024025
****10011012019020021
******13014017018
                        **6070809022023024025
                       ****10011012019020021
*****13014017018
```

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

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

Note: $1 \le N \le 10^8$

Hint: 153isa3-digitnumber, and 153=1^3+5^3+3^3. Sample Input:

*******15016

*******15016

Passed all tests! ✓

153

Sample Output:

true

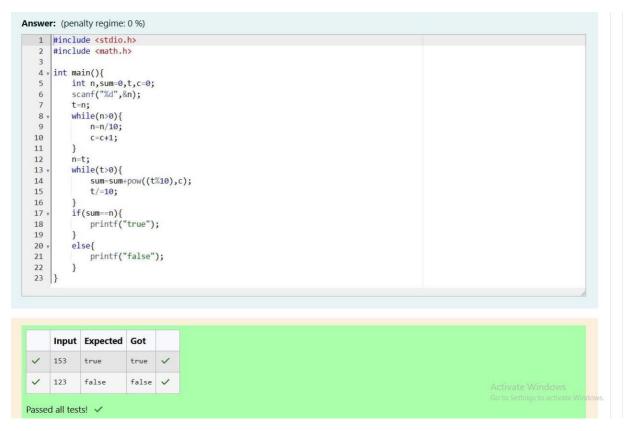
SampleInput: 123
SampleOutput: false

SampleInput:

1634

SampleOutput:

true



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

SampleInput1

32

Sample Output 1

55

SampleInput2

789

Sample Output 2

66066

```
Answer: (penalty regime: 0 %)
     #include <stdio.h>
   4 int rev(int n){
          int reverse=0;
          while(n>0){
          reverse=(reverse*10)+(n%10);
   8
          n=n/10;
   9
  10
          return reverse;
  11 }
  12 - int Pal(int n){
          return rev(n)==n;
  13
  15 v int main(){
          int n;
scanf("%d",&n);
  16
  17
  18 🔻
          while(!Pal(n)){
  19
             int r=rev(n);
  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 programto printthe nth lucky number. Example, 1stlucky 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 notlucky as they have other numbers in it. The program should accept a number 'n' as input and displaythe nth lucky number as output.

```
SampleInput1:
3
Sample Output 1:
33
```

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

