NINETHA NATARAJAN N ECE-D 240801221 **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 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

Sample Output:

WBW

BWB

WBW

WBWBW

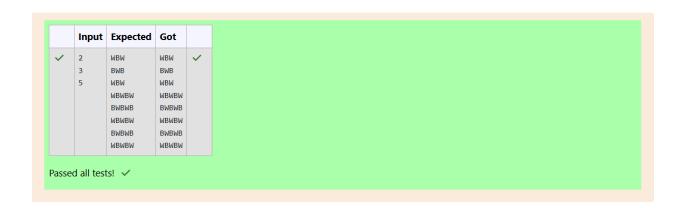
BWBWB

WBWBW

BWBWB

WBWBW

```
Answer: (penalty regime: 0 %)
   1 #include<stdio.h>
    2 v int main(){
    3
            int T,size;
            scanf("%d",&T);
while(T--){
    scanf("%d",&size);
    4
    5
    6
                 for(int i=0;i<size;i++){</pre>
                      for(int j=0;j<size;j++){
   if((i+j)%2==0){</pre>
    8 •
   9 ,
                                printf("W");
   10
   11
  12 •
                           else{
                                printf("B");
   13
   14
   15
                      printf("\n");
   16
   17
   18
             if(T>0){
  19
                 printf("\n");
   20
   21
   22 }
```



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 W

3 B

Sample Output:

WB

BW

BWB

WBW

BWB

```
Answer: (penalty regime: 0 %)

1  | minclude <stdio.h>
2  | int main(){
4  | char s;
5  | int t,n;
6  | scanf("%d,%c",&n,&s);
7  | while(t--){
8  | scanf("%d,%c",&n,&s);
for(int i-0;icn;i+){
10  | for(int j-0;jcn;j+){
11  | if((i+j)%c-0){
12  | printf("%c",s);
13  |
14  |
15  | if(s="\"){
16  | printf("b");
17  |
18  | else{
19  | printf("\");
20  |
21  |
22  |
23  |
24  |
25  |
26  |

27  |
28  |
29  |
20  |
20  |
21  |
22  |
23  |
24  |
25  |
26  |

27  |
28  |
29  |
20  |
20  |
21  |
22  |
23  |
24  |
25  |
26  |
```

Problem Statement 3:

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 3 3 4 5 Sample Output Case #1 10203010011012

6

**4050809

****607

Case #2

1020304017018019020

**50607014015016

****809012013

******10011

Case #3

102030405026027028029030

**6070809022023024025

****10011012019020021

******13014017018

*******15016

```
Answer: (penalty regime: 0 %)
    1 #include<stdio.h>
    2 v int main(){
             int t;
scanf("%d",&t);
for(int x = 1;x <= t;x++){
    printf("Case #%d\n",x);</pre>
    3
    4
    5 ,
    6
                   int n;
scanf("%d",&n);
    7
    8
    9
                   int f=1,b=n*(n+1);
                   for(int i=0;i<n;i++){
   10 ▼
                        for(int k=0;k< 2*i ;k++){
    printf("*");
   11 v
   12
   13
   14
                        printf("%d",f);
   15
                       f++;
for(int j=2;j<= n-i ;j++){
    printf("0%d",f);</pre>
   16
   17 🔻
   18
                             f++;
   19
   20
                        for(int l=b-(n-i)+1;l<=b;l++){
   21 1
                            printf("0%d",1);
   22
   23
                       b-=n-i;
printf("\n");
   24
   25
   26
   27
   28
              return 0;
   29 }
```

	Input	Expected	Got	
~	3	Case #1	Case #1	~
	3	10203010011012	10203010011012	
	4	**4050809	**4050809	
	5	****607	****607	
		Case #2	Case #2	
		1020304017018019020	1020304017018019020	
		**50607014015016	**50607014015016	
		****809012013	****809012013	
		*****10011	*****10011	
		Case #3	Case #3	
		102030405026027028029030	102030405026027028029030	
		**6070809022023024025	**6070809022023024025	
		****10011012019020021	****10011012019020021	
		*****13014017018	*****13014017018	
		******15016	******15016	
Passe	d all test	ts! ✓		
e	d all test	ts! ✓		

Problem Statement 4:

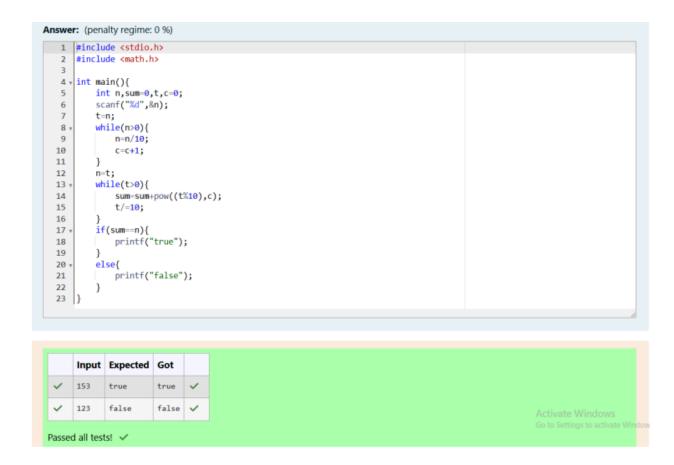
The k-digit number N is an Armstrong number if and only if the k-th power of each digit sums to N. Given a positive integer N, return true if and only if it is an Armstrong number.

Note: 1 <= 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

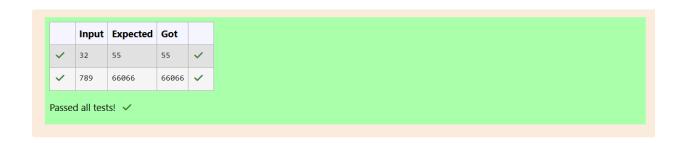


Problem Statement 5:

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

```
Answer: (penalty regime: 0 %)
      #include<stdio.h>
       #include<limits.h>
   3 v long long reversenumber(long long num){
          long long reversed=0;
   4
   5
          while(num>0){
              int digit=num%10;
   6
               if(reversed>(LONG_MAX-digit)/10){
   7
   8
                   return -1;
   9
              reversed=reversed*10+digit;
  10
  11
               num/=10;
  12
          return reversed;
  13
  14
  15 v int ispallindrome(long long num){
          return num==reversenumber(num);
  16
  17
  18
  19 v int main(){
  20
          long long num;
           scanf("%lld",&num);
  21
           while(1){
  22 1
  23
          long long reversed=reversenumber(num);
          num=num+reversed;
  24
               if(ispallindrome(num)){
  25
  26
                  printf("%lld",num);
                   break;
  27
  28
  29
  30
  31 }
```



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:

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

