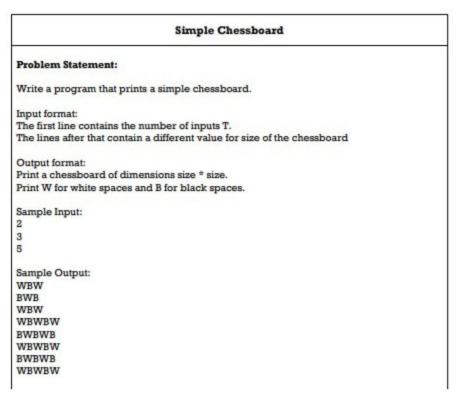
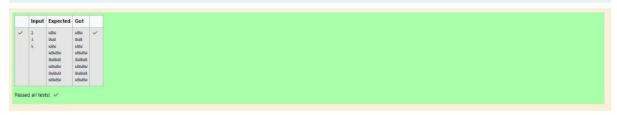
NAME: DHANALAKSHMI.C

REGISTER NUMBER: 241901023

WEEK-05-01







Print Our Own Chessboard

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

```
#include <stdio.h>
int main()
3 +
4
5
6
7
             int T,d,i,i1,i2,o,z;
            char c,s;
scanf("%d",&T);
for (i=0;i<T;i++)
                    scanf("%d %c",&d,&s);
for (i1=0;i1<d;i1++)
10
11 -
                         z=(s=='W')?0:1;
o=(i1%2==z)?0:1;
for (i2=0;i2<d;i2++)
12
13
14
15
                         {
    c=(i2%2==o)?'W':'B';
    printf("%c",c);
17
18
19
                         printf("\n");
20
21
             return 0;
23 }
```

```
Input Expected Got

2 WB WB WB
2 W BW BW
3 B BNB BWB
WBW WBW
BWB BWB
```

Passed all tests! ✓

Pattern Printing

Problem Statement:

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

First line print Case #i where i is the test case number, In the subsequent line, print the pattern

Sample Input

3 4

5

Sample Output Case #1

10203010011012 **4050809 ****607

Case #2

1020304017018019020

50607014015016 **809012013

*****10011

Case #3

102030405026027028029030

**6070809022023024025

****10011012019020021

******13014017018 *******15016

```
#include <stdio.h>
int main()
1
2
3 v
4
5
6
7 v
8
9
               int n,v,p3,c,in,i,i1,i2,t,ti;
scanf("%d",&t);
for(ti=0;ti<t;ti++)</pre>
                     v=0;
scanf("%d",&n);
printf("Case #%d\n",ti+1);
for (i=0;i<n;i++);</pre>
11
12
13
14
15
                             c=0;
if(i>0)
16
17
18
                                     for (i1=0;i1<i;i1++) printf("**");
                              }
for (i1=i;i1<n;i1++)
19
20
21
22
                                   if (i>0) c++;
printf("%d0",++v);
                             }
if(i==0)
23
24
25
26
27
28
29
30
31
32
                                     p3=v+(v*(v-1))+1;
in=p3;
                            }
in=in-c;
p3=in;
for (i2=i;i2<n;i2++)
                                    printf("%d",p3++);
if (i2!=n-1) printf("0");
33
34
35
36
37
38
                             printf("\n");
       }
```

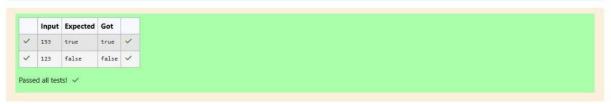
~

WEEK-05-02

```
Status
Started Monday, 23 December 2024, 5:33 PM
Completed Duration
Duration

Finished
Monday, 23 December 2024, 5:33 PM
Tuesday, 26 November 2024, 9:31 PM
26 days 20 hours
```

```
Armstrong Number
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
```



Reverse and Add Until Get a Palindrome

Problem Statement:

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

Constraints 1<=num<=999999999

Sample Input 1

32

Sample Output 1

55

Sample Input 2

789

Sample Output 2

66066

```
1 | #include<stdio.h>
2 | int main()
3 * {
4 | int rn,n,nt=0,i=0;
5 | scanf("%d",8n);
6 | do
7 * {
8 | nt=n;rn=0;
9 | while(n!=0)
10 * {
11 | rn=rn*10+n%10;
12 | n=n/10;
13 | }
14 | n=nt+rn;
15 | i++;
16 | }
17 | while(rn!=nt||i=-1);
18 | printf("%d",rn);
19 | return 0;
20 | }
```

Input	Expected	Got	
32	55	55	~
789	66066	66066	~

Lucky Number

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

