Nested Loops - while and for, Jumps in Loops

Ex.No.: **Date:**

SimpleChessboard

ProblemStatement:

Writeaprogramthatprintsasimplechessboard.

Inputformat:
ThefirstlinecontainsthenumberofinputsT.
The linesafterthatcontainadifferent valueforsizeofthechessboard

Outputformat:
Print a chessboard of dimensions size * size.
Print W for white spaces and B for black spaces.

SampleInput:
2
3
5

Sample Output:
WBW
BWB
WBW
WBWBW
BWBWB
WBWBW
BWBWB
WBWBW

Program:

```
#include<stdio.h>
int main()
{
    int t,size,i,j;
    scanf("%d",&t);
    while(t--)
    {   scanf("%d",&size);
        for(i=0;i<size;i++)
        {
            for(j=0;j<size;j++)
            {
                printf("%c",(i+j)%2==0?'W

            }printf("\n");
        }
    }
}
```

	Input	Expected	Got	
✓	2 3 5	WBW BWB WBW WBWBW BWBWB WBWBW BWBWB WBWBW	WBW BWB WBW WBWBW BWBWB WBWBW BWBWB WBWBW	✓

Passed all tests! ✓

Ex.No.: **Date:**

PrintOurOwnChessboard

ProblemStatement:

Let's printachessboard!

Writeaprogramthattakesinput:
The firstline contains T, the numberoftestcases
EachtestcasecontainsanintegerNandalsothestartingcharacterofthechessboard

OutputFormat
Printthechessboardasperthegivenexamples

SampleInput:
2
2W
3B

Sample Output:
WB
BW
BWB
WBW
BWB

Program:

```c
#include<stdio.h>
int main()
{
    int t,size,i,j;
    char c,x,y;
    scanf("%d",&t);
    while(t--)
    {
        scanf("%d %c ",&size,&c);
        if(c=='W'){
            x='W';y='B';
        }
        else{
            x='B';y='W';
        }

        for(i=0;i<size;i++)
        {
            for(j=0;j<size;j++)
            {
                printf("%c",(i+j)%2==0?x:y
            }printf("\n");


        }
    }
}
```

	Input	Expected	Got	
✓	2 2 W 3 B	WB BW BWB WBW BWB	WB BW BWB WBW BWB	✓

Passed all tests! ✓

Ex.No.: **Date:**

PatternPrinting

ProblemStatement:

DecodethelogicandprintthePattern thatcorrespondstogiveninput.

If N=3 then patternwillbe:
10203010011012
**4050809
****607
IfN=4, thenpattern will be:
1020304017018019020
**50607014015016
****809012013
******10011

Constraints: 2 <= N <= 100
Input Format
FirstlinecontainsT,thenumberoftestcases,eachtestcasecontainsasingleintegerN

OutputFormat
FirstlineprintCase#iwhereiisthetestcasenumber,Inthesubsequentline,printthe pattern

SampleInput
3
3
4
5

SampleOutput
Case #1
10203010011012
**4050809
****607
Case#2
1020304017018019020
**50607014015016
****809012013
******10011
Case#3
10203040502602702802 9030
**6070809022023024025
****10011012019020021
******13014017018
********15016

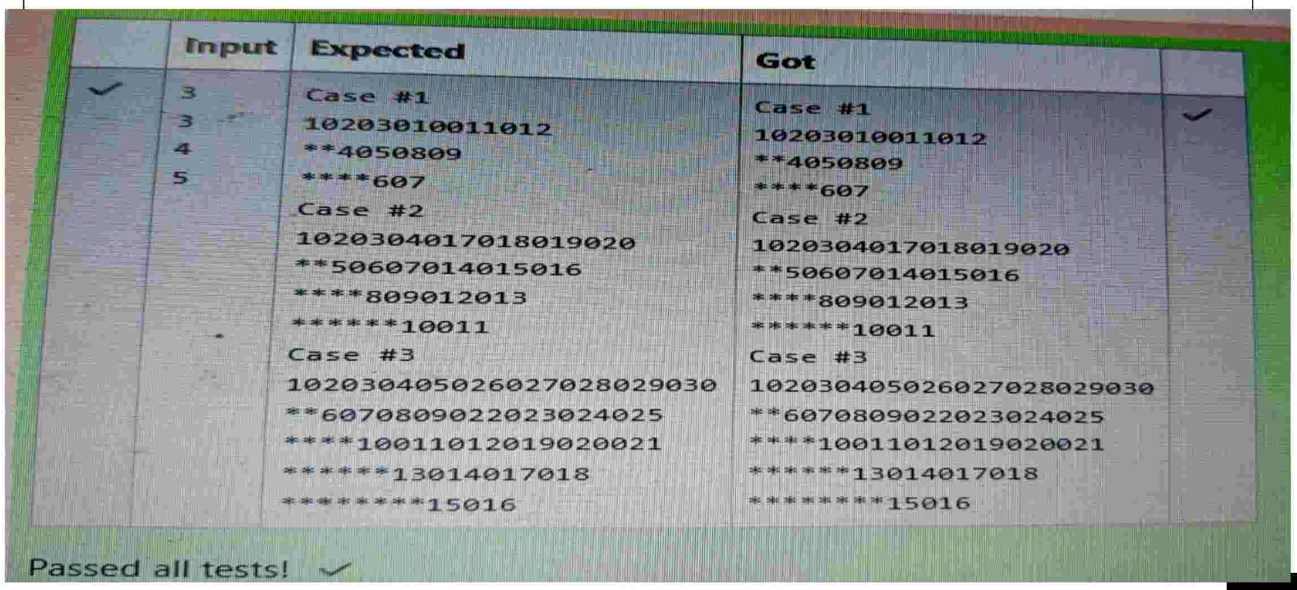
Program:

```c
#include<stdio.h>
int main()
{
    int n,v,p3,c,in,i1,i2,t,ti,i;
    scanf("%d",&t);
    for(ti=0;ti<t;ti++)
    {
        v=0;
        scanf("%d",&n);
        printf("Case #%d\n",ti+1);
        for(i=0;i<n;i++)
        {
            c=0;
            if(i>0)
            {
                for(i1=0;i1<i;i1++)
                printf("**");

            }
        for(i1=i;i1<n;i1++)
        {
            if(i>0) c++;
            printf("%d0",++v);

        }
        if(i==0){
            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");
        }printf("\n");


            }
        }
    }
```



Ex.No.: **Date:**

ArmstrongNumber

ProblemStatement:

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

GivenapositiveintegerN,returntrueifandonlyifitisanArmstrongnumber.

Note:1<=N<=10^8

Hint:153is a3-digitnumber,and153= 1^3+5^3+3^3.

SampleInput:
153

Sample Output:
true

Sample **Input**:
123

Sample **Output**:
false

Sample **Input**:
1634

Sample **Output**:
true

Program:

```c
#include<stdio.h>
#include<math.h>
int main()
{
    int n;
    scanf("%d",&n);
    int x=0,n2=n;
    while(n2!=0)
    {
        x++;
        n2=n2/10;
    }
    int sum=0;
    int n3=n,n4;
    while(n3!=0)
    {
        n4=n3%10;
        sum=sum+pow(n4,x);
        n3=n3/10;

    }
    if(n==sum)
     printf("true");
    else
     printf("false");


}
```

	Input	Expected	Got	
✓	153	true	true	✓
✓	123	false	false	✓

Passed all tests! ✓

Ex.No.: **Date:**

ReverseandAddUntilGetaPalindrome

ProblemStatement:

Takeanumber,reverseitandadditotheoriginalnumberuntiltheobtainednumberis a palindrome.

Constraints
1<=num<=99999999

SampleInput1
32

SampleOutput1
55

SampleInput2
789

SampleOutput2
66066

Program:

```c
#include<stdio.h>
int main()
{
    int rn,n,nt=0,i=0;
    scanf("%d",&n);
    do{
        nt=n;rn=0;
        while(n!=0)
        {
            rn=rn*10+n%10;
            n=n/10;
        }
        n=nt+rn;
        i++;
    }
    while((rn!=nt)||i==1);
    printf("%d",rn);
    return 0;
}
```

	Input	Expected	Got	
✓	32	55	55	✓
✓	789	66066	66066	✓

Passed all tests! ✓

Ex.No.: **Date:**

LuckyNumber

ProblemStatement:

Anumberisconsideredluckyifitcontainseither3or4or3and4bothinit.Writea programtoprintthenthluckynumber.Example,1stluckynumberis3,and2ndlucky numberis4and3rdluckynumberis33and4thluckynumberis34andsoon.Notethat 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.

SampleInput1:
3

SampleOutput1:
33

Program:

```c
#include<stdio.h>
int main()
{
    int n=1,i=0,nt,co=0,e;
    scanf("%d",&e);
    while(i<e)
    {
        nt=n;
        while(nt!=0)
        {
            co=0;
            if(nt%10!=3 && nt%10!=4)
            {
                co=1;
                break;

            }
            nt=nt/10;
        }
            if(co==0)
            {
                i++;
            }
            n++;
        }
        printf("%d",--n);

}
```

	Input	Expected	Got	
✓	34	33344	33344	✓

Passed all tests! ✓