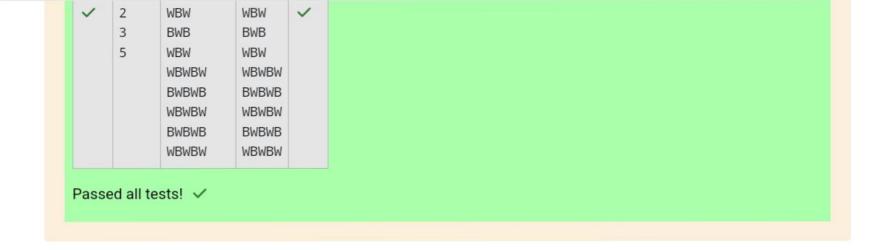
PROGRAMMING USING C WEEK 5 (NESTED LOOPS)

NAME: TARANYA.M

DEPT:IT-E

ROLL NO:241001285



Question **2**Correct
Marked out of 5.00

Figure Flag question

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 / Output

```
3 B
```

Output:

WB

BW

BWB

WBW

BWB

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
2 *
   int main(){
 3
        int T,d,i,i1,i2,o,z;
        char c,s;
 4
 5
        scanf("%d",&T);
 6
        for(i=0;i<T;i++)
 7 *
 8
            scanf("%d %c",&d,&s);
 9
            for(i1=0;i1<d;i1++)
10 *
11
                 z=(s=='W') ? 0:1;
12
                 o=(i1\%2==z) ? 0:1;
13
                 for(i2=0;i2<d;i2++)
14 ₹
15
                     c=(i2\%2==o) ?'W':'B';
16
                     printf("%c",c);
17
18
                 printf("\n");
```

```
15 | c=(i2%2==o) ?'W':'B';
16 | printf("%c",c);
17 | }
18 | printf("\n");
19 | }
20 | }
21 | return 0;
22 |}
```

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

Passed all tests! 🗸

Question **3**Correct

Marked out of

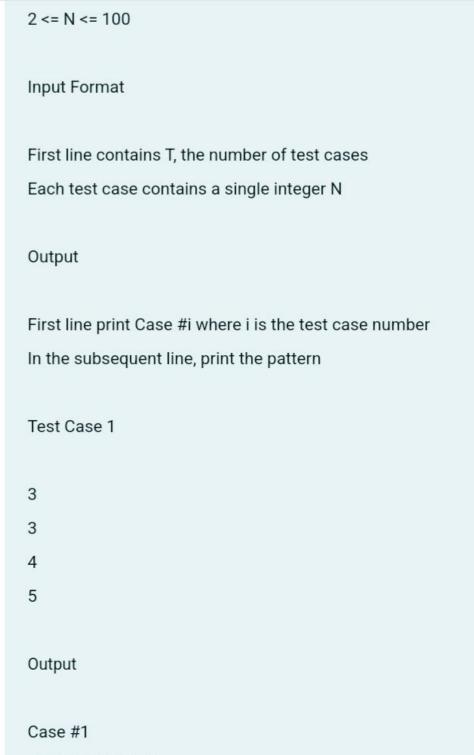
7.00

Flag question

Decode the logic and print the Pattern that corresponds to given input.

If N= 3

then pattern will be:



```
Case #3
102030405026027028029030
**6070809022023024025
****10011012019020021
*****13014017018
*****15016
Answer: (penalty regime: 0 %)
       #include<stdio.h>
   2 *
       int main(){
           int v,c=0;
   4
           scanf("%d",&v);
   5 *
           while(v!=0){
   6
               C++;
               int a;
   8
               scanf("%d",&a);
               int s1=10, s2=(a*a*10)+10;
   9
   10
               printf("Case #%d\n",c);
  11 v
                for(int i=0; i<a; i++){
                    for(int j=0;j<i;j++){
  12 *
  13
                        printf("**");
  14
  15 *
                    for(int j=0; j<a-i; j++){
  16
                        printf("%d",s1);
  17
                        s1+=10;
  18
```

printf("%d",((s2+(j*10))/10));

for(int j=0;j<a-i;j++){

1001

 $if((j+1)==(a-i)){$

^^^^ 10011

19 .

20 *

21

```
22
                    else{
23 *
                         printf("%d",(s2+(j*10)));
24
25
26
27
                s2 -= (a-i)*10;
28
                s2+=10;
                printf("\n");
29
30
31
32
33
```

	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

Passed all tests! <

Question 1 Correct	The k-digit number N is an Armstrong number if and only if the k-th power of each digit sums to N.
Marked out of 3.00	Given a positive integer N, return true if and only if it is an Armstrong number.
	Example 1:
	Input:
	153
	Output:
	true
	Explanation:
	153 is a 3-digit number, and 153 = 1^3 + 5^3 + 3^3.

false	
Explanation:	
123 is a 3-digit number, and 123 != 1^3 + 2^3 + 3^3 = 36.	
Example 3:	
Input:	
1634	
Output:	
true	
Note:	
1 <= N <= 10^8	
Answer: (penalty regime: 0 %)	
1 #include <stdio.h></stdio.h>	
2 #include <math.h></math.h>	

```
Answer: (penalty regime: 0 %)
      #include<stdio.h>
      #include<math.h>
   3 v int main(){
   4
           int n;
           scanf("%d",&n);
   5
   6
           int x=0, n2=n;
   7
           while(n2!=0)
   8 *
   9
               X++;
  10
               n2=n2/10;
  11
  12
           int sum=0;
  13
           int n3=n, n4;
           while(n3!=0)
  14
  15 *
  16
               n4=n3%10;
  17
               sum=sum+pow(n4,x);
  18
               n3=n3/10;
  19
  20
           if(n==sum)
  21 *
               printf("true");
  22
  23
           else{
  24 *
  25
               printf("false");
  26
  27
           return 0;
  28
```

Г		Input	Expected	Got	
	~	153	true	true	~

	Input	Expected	Got	
~	153	true	true	~
~	123	false	false	~

Passed all tests! 🗸

Question **2**

Correct
Marked out of
5.00

Flag question

```
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 66066
```

Answer: (penalty regime: 0 %)

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

	Input	Expected	Got	
~	32	55	55	~
~	789	66066	66066	~

Passed all tests! <

Question **3**Correct
Marked out of 7.00

Flag question

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:

```
Sample Input 2:

34

Sample Output 2:
```

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
 2 v
    int main(){
 3
        int n=1,i=0,nt,co=0,e;
 4
        scanf("%d",&e);
 5
        while(i<e)</pre>
 6 *
 7
             nt=n;
 8
             while(nt!=0)
 9 *
10
                 co=0;
11
                 if(nt%10!=3&&nt%10!=4)
12 *
13
                      co=1;
                      break;
14
15
```

```
8
            while(nt!=0)
 9
10
                 co=0;
11
                 if(nt%10!=3&&nt%10!=4)
12 *
13
                     co=1;
14
                     break;
15
16
                 nt=nt/10;
17
18
             if(co==0)
19 +
20
                 i++;
21
22
            n++;
23
24
        printf("%d",--n);
25
        return 0;
26
```

	Input	Expected	Got	
--	-------	----------	-----	--

Passed all tests! <