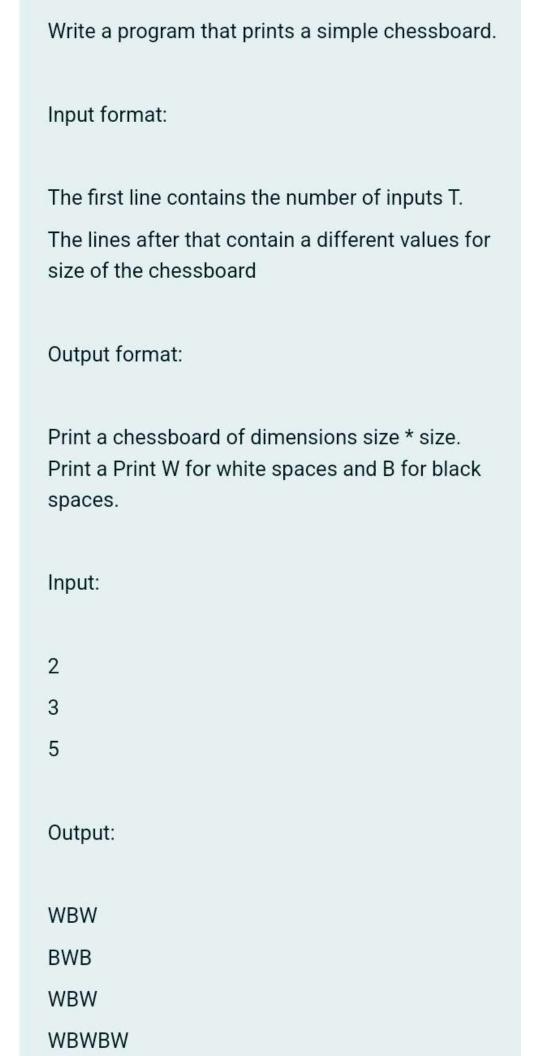
## Your attempts

## Attempt 1

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Friday, 22 November 2024, 2:17 PM
Duration	31 days 3 hours

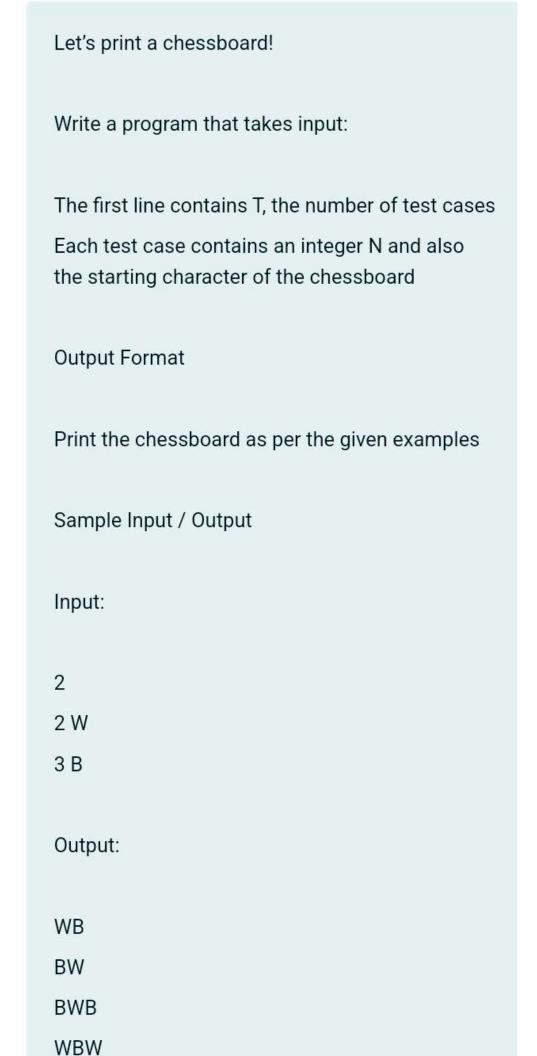
Review



## Answer: (penalty regime: 0 %)

```
#include<stdio.h>
 1
 2
    int main()
 3
    {
 4
         int T;
 5
         scanf("%d",&T);
 6
         for(int i=0;i<T;i++){
 7
              int size;
 8
              scanf("%d",&size);
 9
              for(int j=0;j<size;j++).</pre>
10 *
                  for(int k=0;k<size;
11
                       if((j+k)\%2==0)
12 *
13
                            printf("W")
14
15 w
                       else{
16
                           printf("B")
17
18
19
                  printf("\n");
20
21
22
23
         return 0;
24
    }
```

	Input	Expected	Got	
/	2	WBW	WBW	~
	3	BWB	BWB	
	5	WBW	WBW	
		WBWBW	WBWBW	
		BWBWB	BWBWB	
		WBWBW	WBWBW	
		BWBWB	BWBWB	
		WBWBW	WBWBW	



```
#include<stdio.h>
 1
    int main()
 3 ₹
    {
 4
         int T,d,i,i1,i2,o,z;
 5
         char c,s;
 6
         scanf("%d",&T);
 7
         for(i=0;i<T;i++)
 8 *
 9
             scanf("%d %c",&d,&s);
10 v
             for(i1=0;i1<d;i1++){
11
                  z=(s=='W')?0:1;
12
                  o=(i1\%2==z)?0:1;
13 v
                  for(i2=0;i2<d;i2++).
14
                      c=(i2\%2==o)?'W'
15
                      printf("%c",c);
16
                  printf("\n");
17
18
19
20
        return 0;
21
    }
```

```
lude<stdio.h>
    nain()
 3 🔻
 4
    nt T,d,i,i1,i2,o,z;
 5
    har c,s;
 6
    canf("%d",&T);
 7
    or(i=0;i<T;i++)
 8 *
 9
        scanf("%d %c",&d,&s);
10 v
        for(i1=0;i1<d;i1++){
11
            z=(s=='W')?0:1;
12
            o=(i1\%2==z)?0:1;
13 *
            for(i2=0;i2<d;i2++){
14
                c=(i2\%2==o)?'W':'B';
15
                printf("%c",c);
16
            printf("\n");
17
18
19
    eturn 0;
20
21
```

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

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

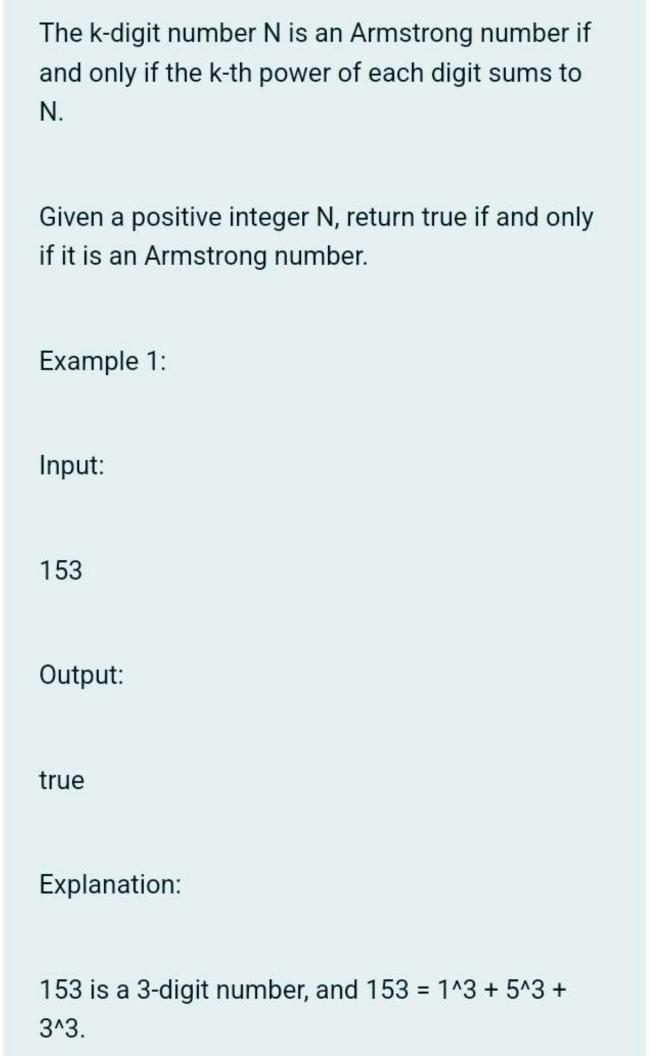
```
input Format
First line contains T, the number of test cases
Each test case contains a single integer N
Output
First line print Case #i where i is the test case
number
In the subsequent line, print the pattern
Test Case 1
3
3
4
5
Output
Case #1
10203010011012
**4050809
****607
Case #2
1020304017018019020
**50607014015016
```

```
1
    #include<stdio.h>
 2
    int main()
 3 🔻
    {
 4
           int v,c=0;
 5
           scanf("%d",&v);
 6
          while(v!=0)
 7 *
           {
 8
               C++;
               int a;
 9
10
               scanf("%d",&a);
11
               int s1=10, s2=(a*a*10)+1
12
               printf("Case #%d\n",c)
13
               for(int i=0;i<a;i++)
14 *
               {
15
                    for(int j=0;j<i;j+-</pre>
16 *
                    {
17
                        printf("**");
18
                    }
19
                    for(int j=0;j<a-i;j</pre>
20 *
                    {
21
                         printf("%d",s1
22
                         s1+=10;
23
                    }
                    for(int j=0;j<a-i;)</pre>
24
25 *
                    {
26
                         if((j+1)==(a-i
27 *
                         {
28
                            printf("%d'
29
                         }
30 *
                         else{
                              printf("%d'
31
                         }
32
                         }
33
34
                         s2 - = (a - i) * 10;
35
                         s2+=10;
36
                         printf("\n");
37
                    }
38
                    V--;
               }
39
           }
40
41
```

```
1
 2
 3 v
 4
 5
 6
 7 v
 8
 9
10
    &a);
11
    2=(a*a*10)+10;
    e #%d\n",c);
12
    ;i<a;i++)
13
14 ▼
     j=<mark>0</mark>;j<i;j++)
15
16 v
    ntf("**");
17
18
     j=0;j<a-i;j++)
19
20 ▼
    ntf("%d",s1);
21
22
    =10;
23
     j=<mark>0</mark>;j<a-i;j++)
24
25 *
    (j+1)==(a-i)
26
27 *
     printf("%d",((s2+(j*10)))/10);
28
29
30 v e{
31
     printf("%d",(s2+(j*10)));
32
33
    =(a-i)*10;
34
35
    =10;
36
    ntf("\n");
37
38
39
40
41
```

	Input	Expected	Go
~	3	Case #1	Ca
	3	10203010011012	10
	4	**4050809	**
	5	****607	**
		Case #2	Ca
		1020304017018019020	10
		**50607014015016	**
		****809012013	**
		*****10011	**
		Case #3	Ca
		102030405026027028029030	10
		**6070809022023024025	**
		****10011012019020021	**
		*****13014017018	**
		******15016	**

	Got	
	Case #1	~
	10203010011012	
	**4050809	
	****607	
	Case #2	
020	1020304017018019020	
	**50607014015016	
	****809012013	
	*****10011	
	Case #3	
28029030	102030405026027028029030	
24025	**6070809022023024025	
20021	****10011012019020021	
8	*****13014017018	
	******15016	



Input:
123
Output:
false
Explanation:
123 is a 3-digit number, and 123 != 1^3 + 2^3 + 3^3 = 36.
Example 3:
Input:
1634
Output:
true

```
#include<stdio.h>
    #include<math.h>
 3
    int main()
 4
    {
 5
         int n,d,nt,AN,i;
 6
         scanf("%d",&n);
 7
         nt=n;
 8
         for(d=0;nt!=0;d++){
 9
             nt=nt/10;
10
11
         nt=n;
12
         for(i=0;i<d;i++)
13 *
14
             AN=AN+pow(nt%10,d);
15
             nt=nt/10;
16
         if(AN==n)
17
18 *
19
             printf("true");
20
         else{
21 *
22
             printf("false");
23
24
         return 0;
25
    }
```

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

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

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

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

Explanation:

Here the lucky numbers are 3, 4, 33, 34., and the 3rd lucky number is 33.

Sample Input 2:

Explanation:

Here the lucky numbers are 3, 4, 33, 34., and the 3rd lucky number is 33.

Sample Input 2:

34

Sample Output 2:

33344

## Answer: (penalty regime: 0 %)

```
#include<stdio.h>
 2
    int cont(int a)
 3 *
    {
 4
         int c=a;
 5
         while(c!=0)
 6 *
         {
 7
              int d=c%10;
 8
              if(d!=3 && d!=4)
 9
                 return 0;
10
              c/=10;
11
         return 1;
12
13
    }
    int main()
14
15 *
    {
16
         int a,b=0;
         scanf("%d",&a);
17
18
         while(a!=0)
19 *
         {
20
             b++;
21
              if(cont(b))
22 *
              {
23
                  a--;
24
25
26
27
        printf("%d",b);
28
        return 0;
29
    }
```

	Input	Expected	Got	
~	34	33344	33344	~