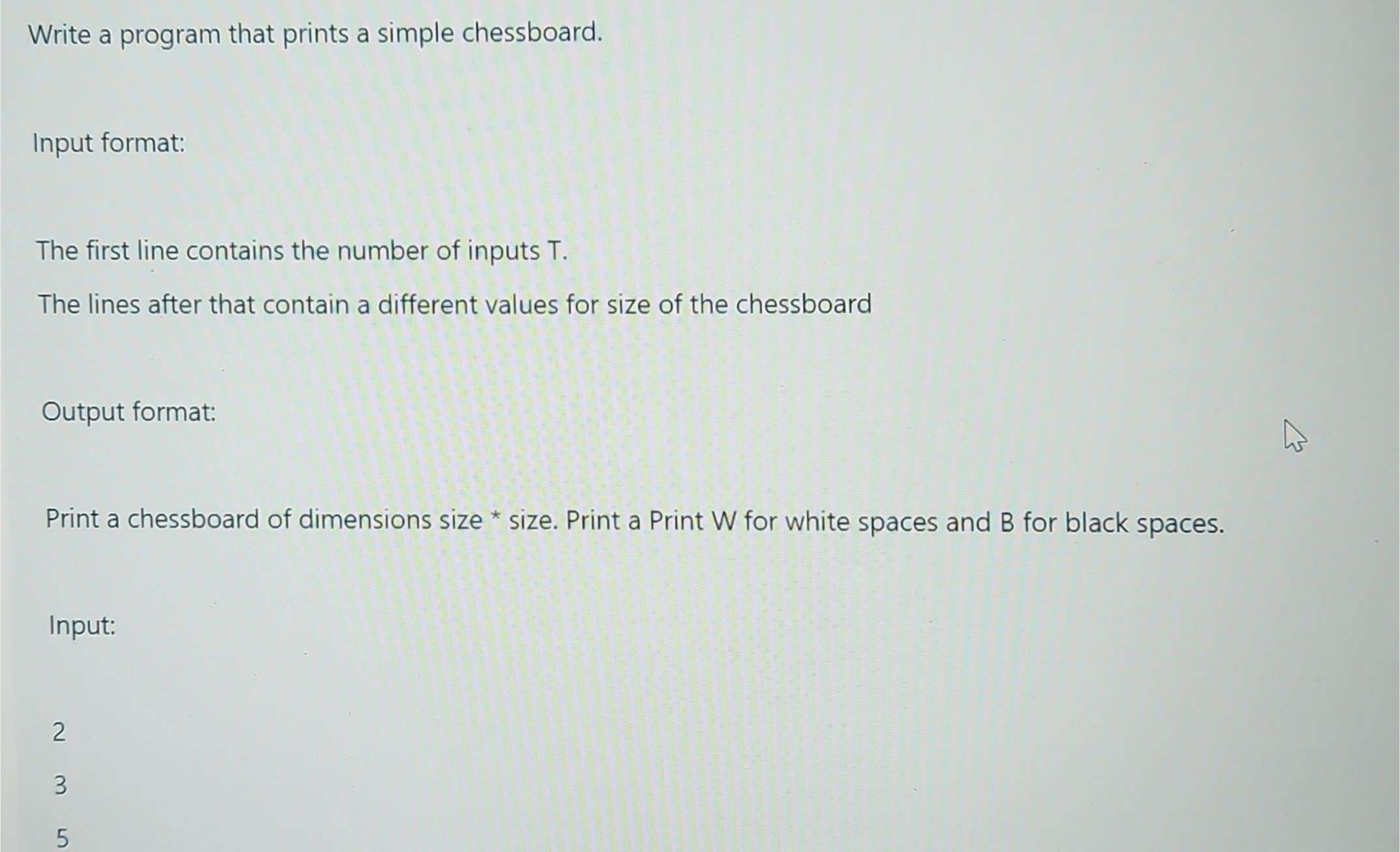


Write a program that prints a simple chessboard.

Input format:

The first line contains the number of inputs T.

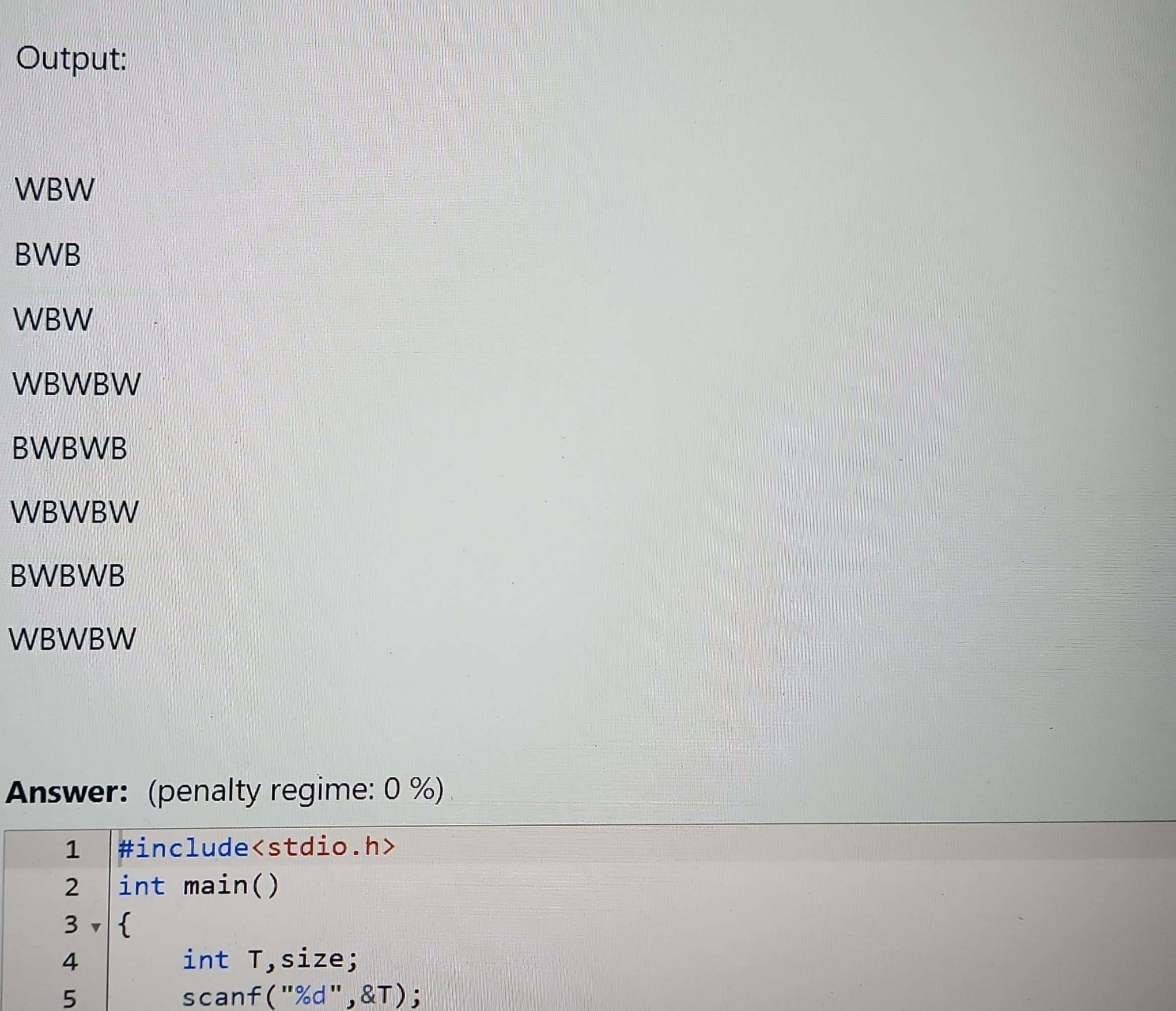
The lines after that contain a different values for size of the chessboard

Output format:

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

Input:

```
2
3
5
```

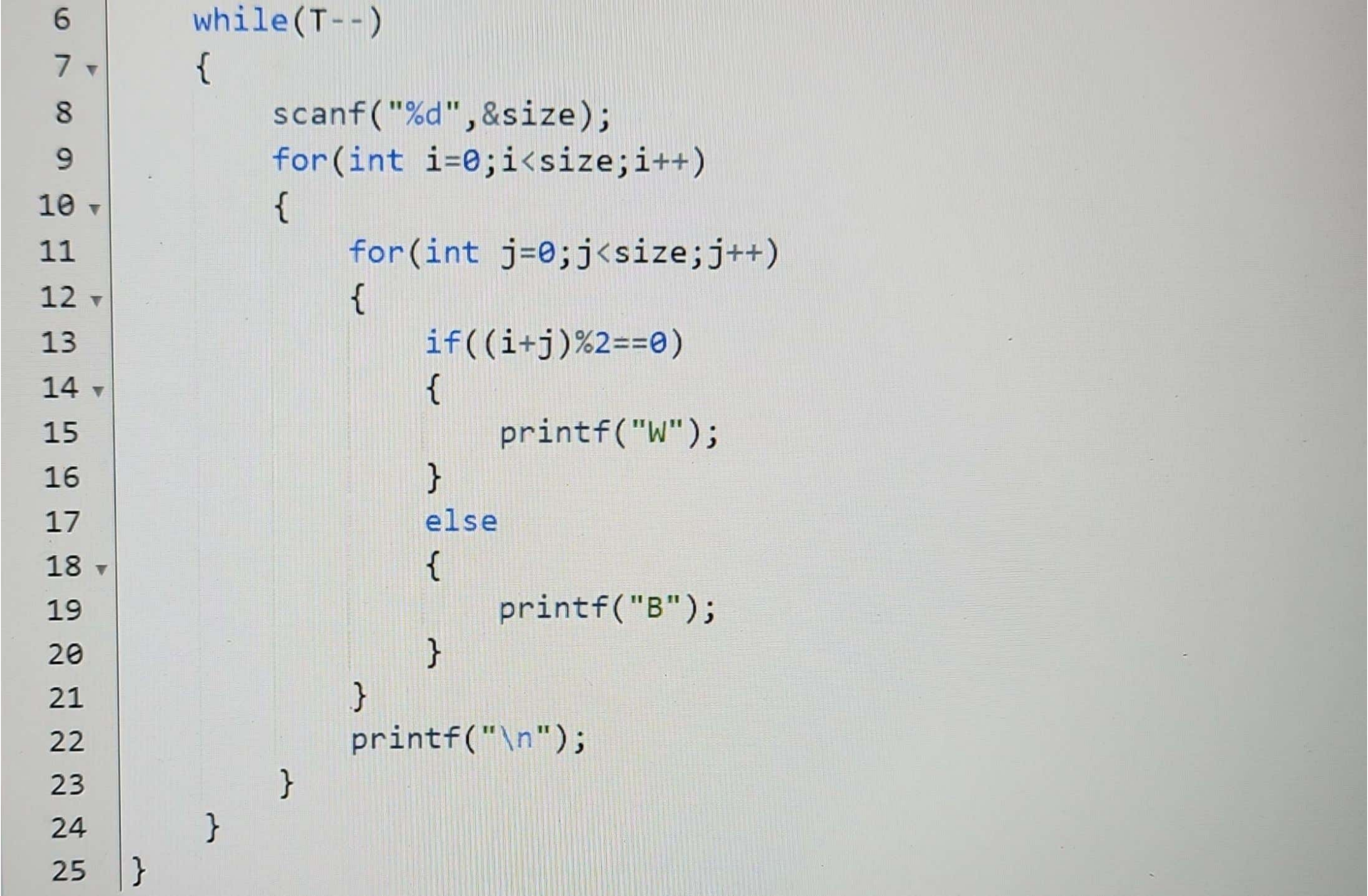


Output:

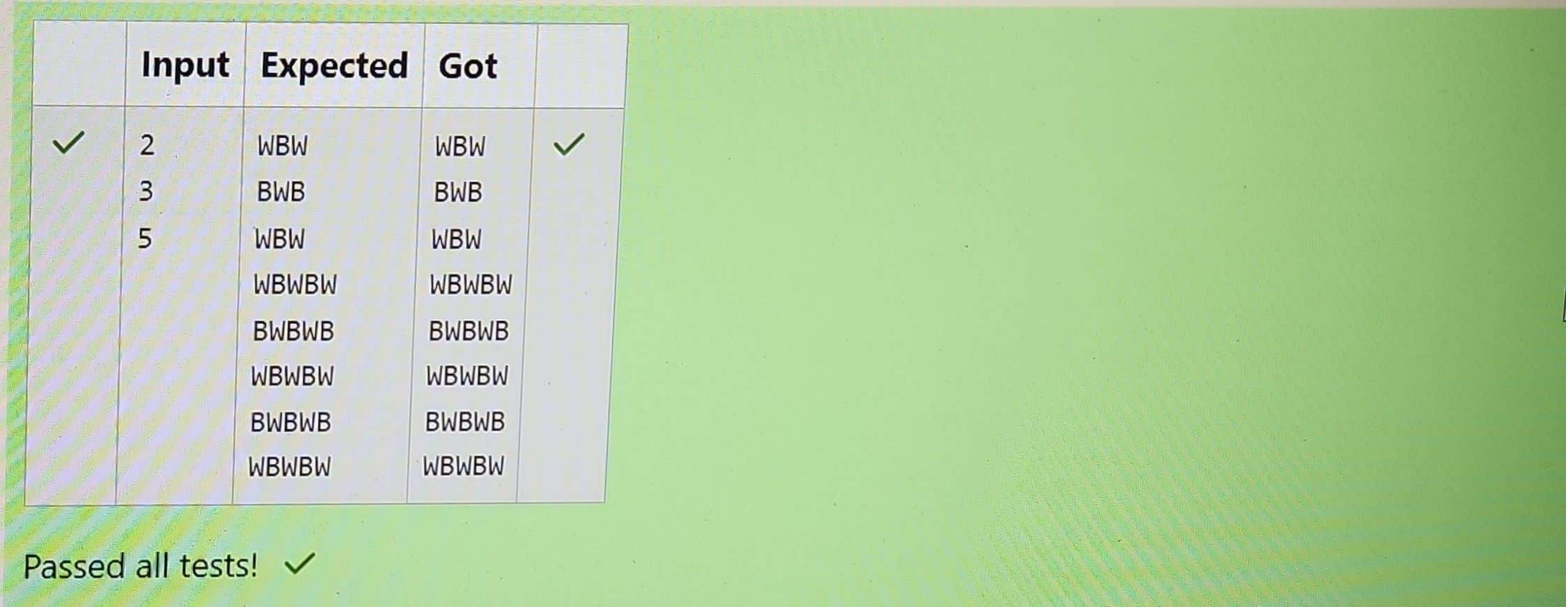
```
WBW
BWB
WBW
WBWBW
BWBWB
WBWBW
BWBWB
WBWBW
```

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int T,size;
    scanf("%d",&T);
```



```
    while(T--)
    {
        scanf("%d",&size);
        for(int i=0;i<size;i++)
        {
            for(int j=0;j<size;j++)
            {
                if((i+j)%2==0)
                {
                    printf("W");
                }
                else
                {
                    printf("B");
                }
            }
            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! ✓

Let's print a chessboard!

Write a program that takes input:

The first line contains T, the number of test cases

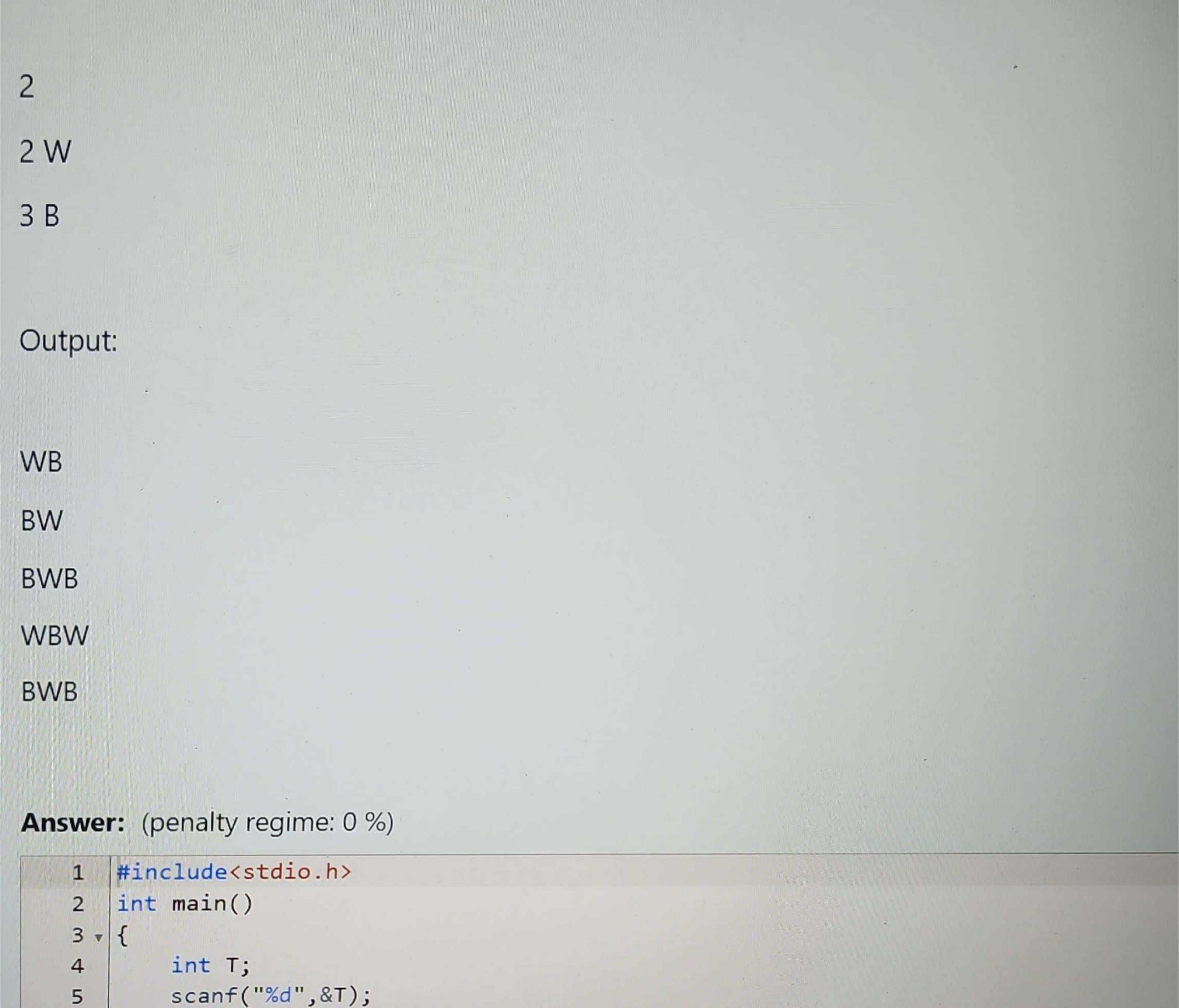
Each test case contains an integer N and also the starting character of the chessboard

Output Format

Print the chessboard as per the given examples

Sample Input / Output

Input:



2

2 W

3 B

Output:

WB

BW

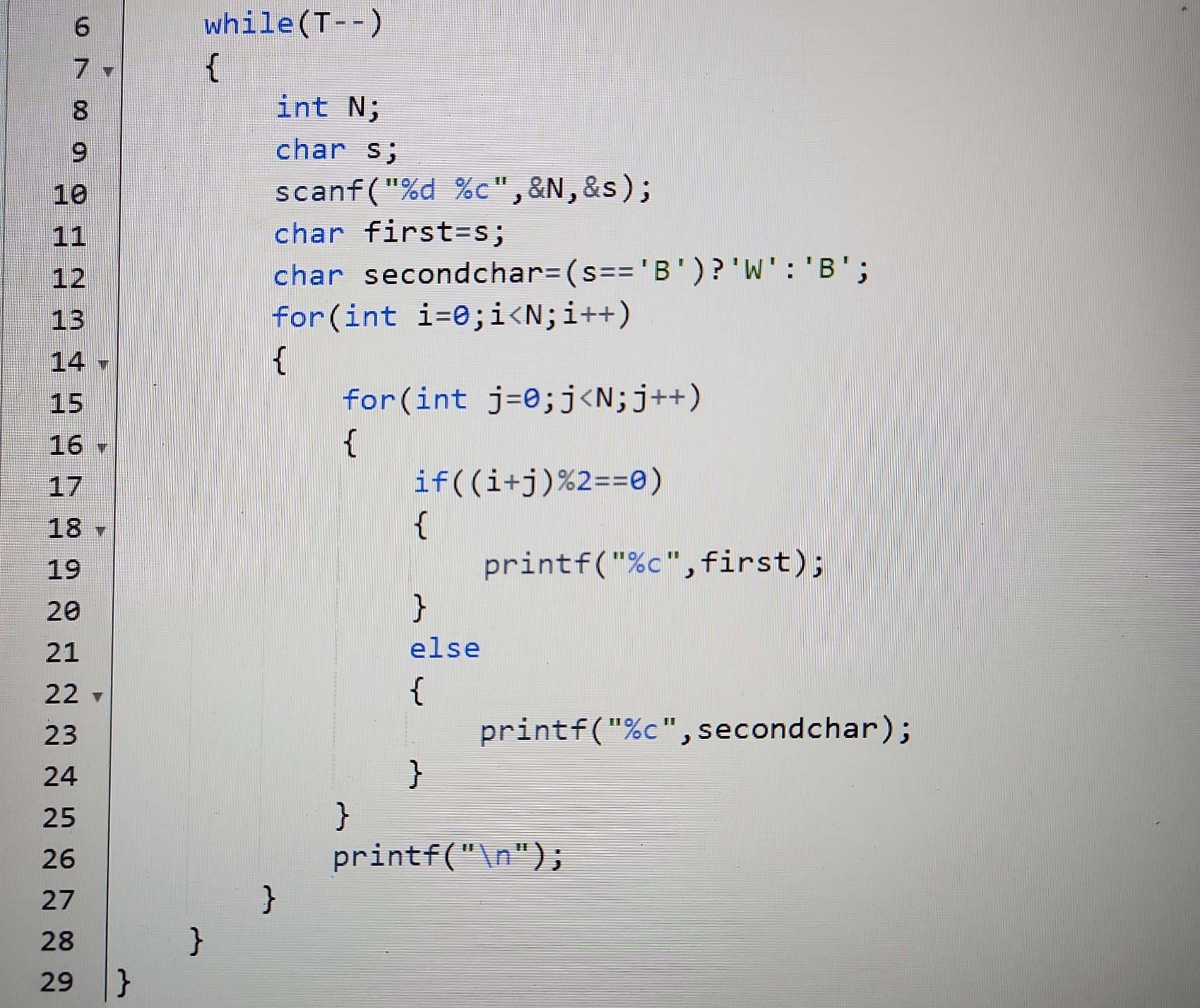
BWB

WBW

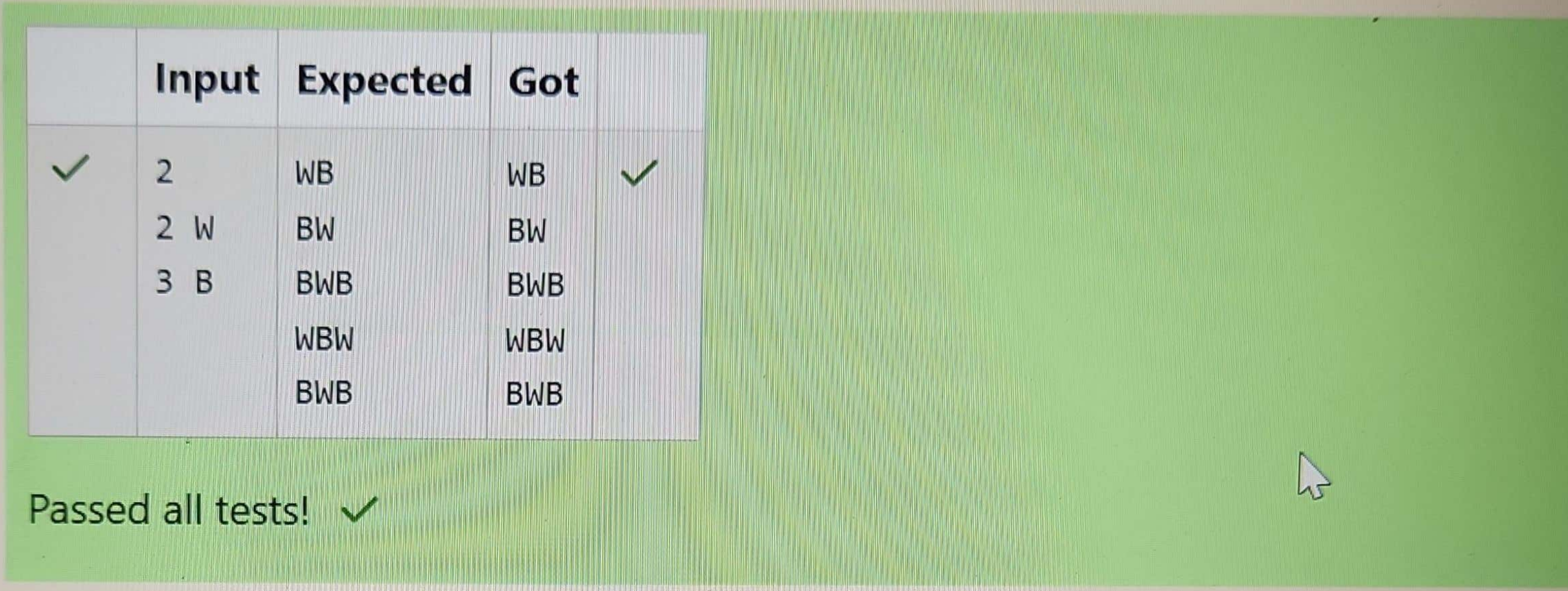
BWB

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int T;
    scanf("%d",&T);
```



```
    while(T--)
    {
        int N;
        char s;
        scanf("%d %c",&N,&s);
        char first=s;
        char secondchar=(s=='B')?'W':'B';
        for(int i=0;i<N;i++)
        {
            for(int j=0;j<N;j++)
            {
                if((i+j)%2==0)
                {
                    printf("%c",first);
                }
                else
                {
                    printf("%c",secondchar);
                }
            }
            printf("\n");
        }
    }
}
```



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

Passed all tests! ✓

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:

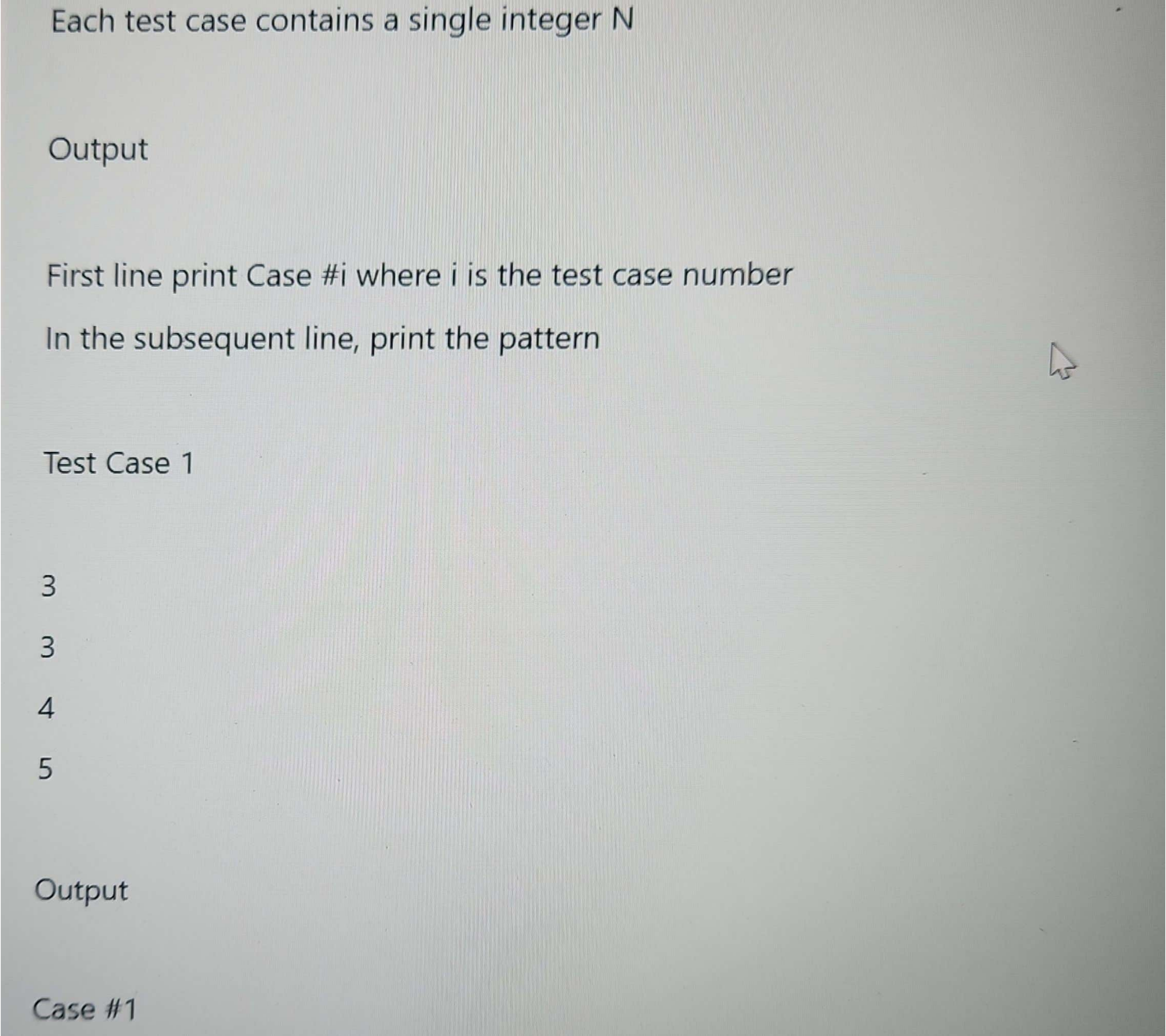
```
10203040170180019020

**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

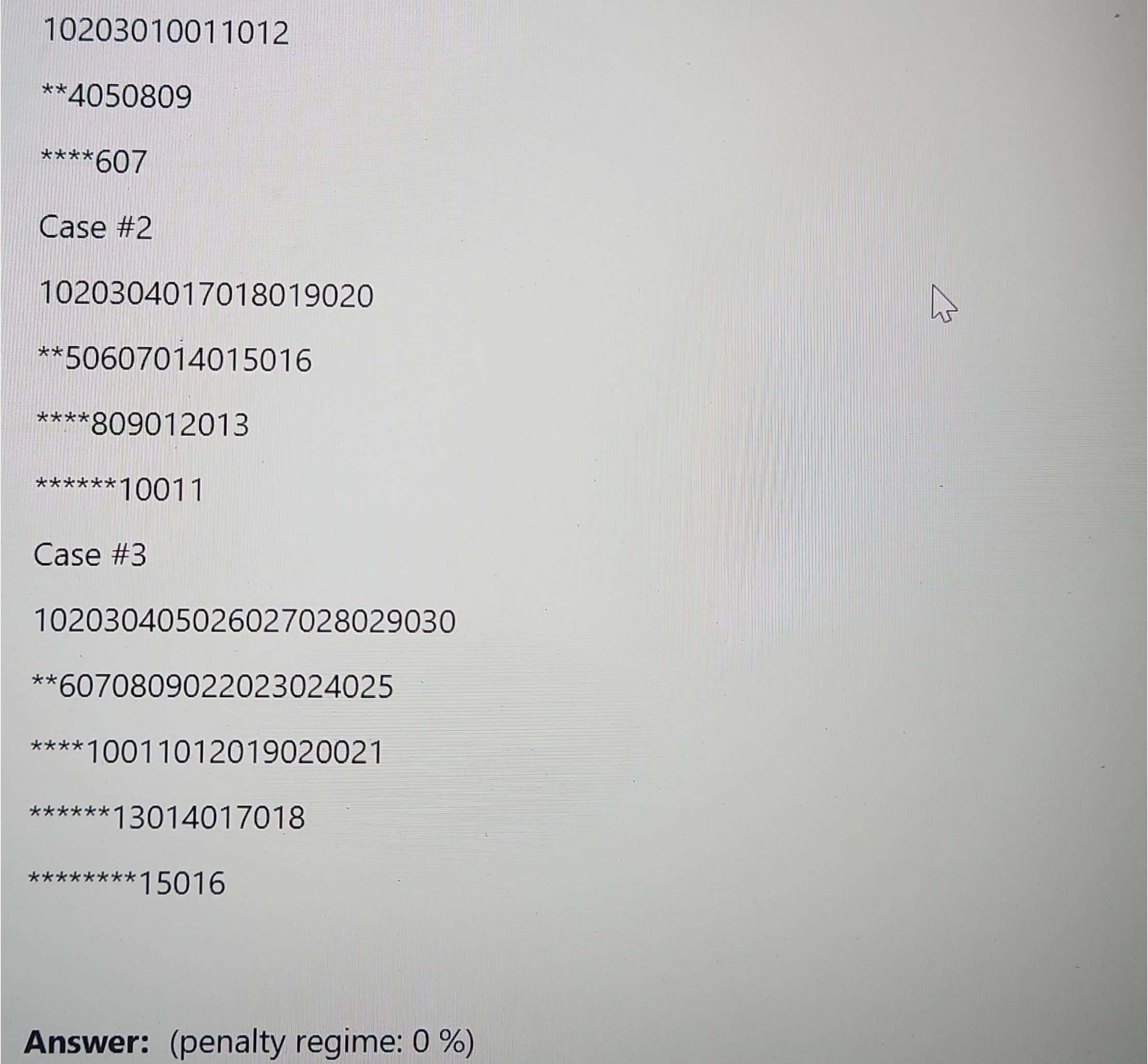
In the subsequent line, print the pattern

Test Case 1

```
3
3
4
5
```

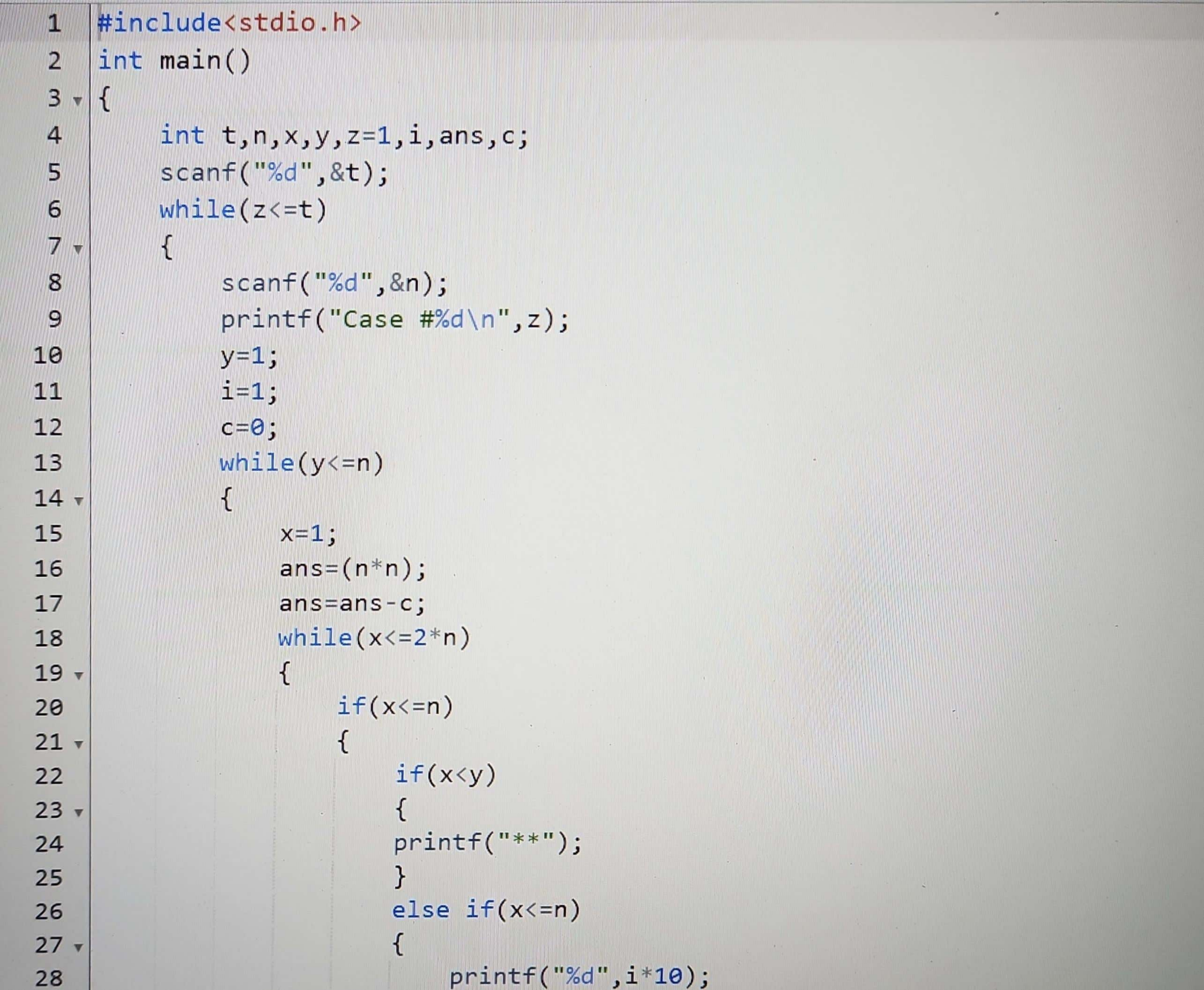
Output

```
Case #1
```

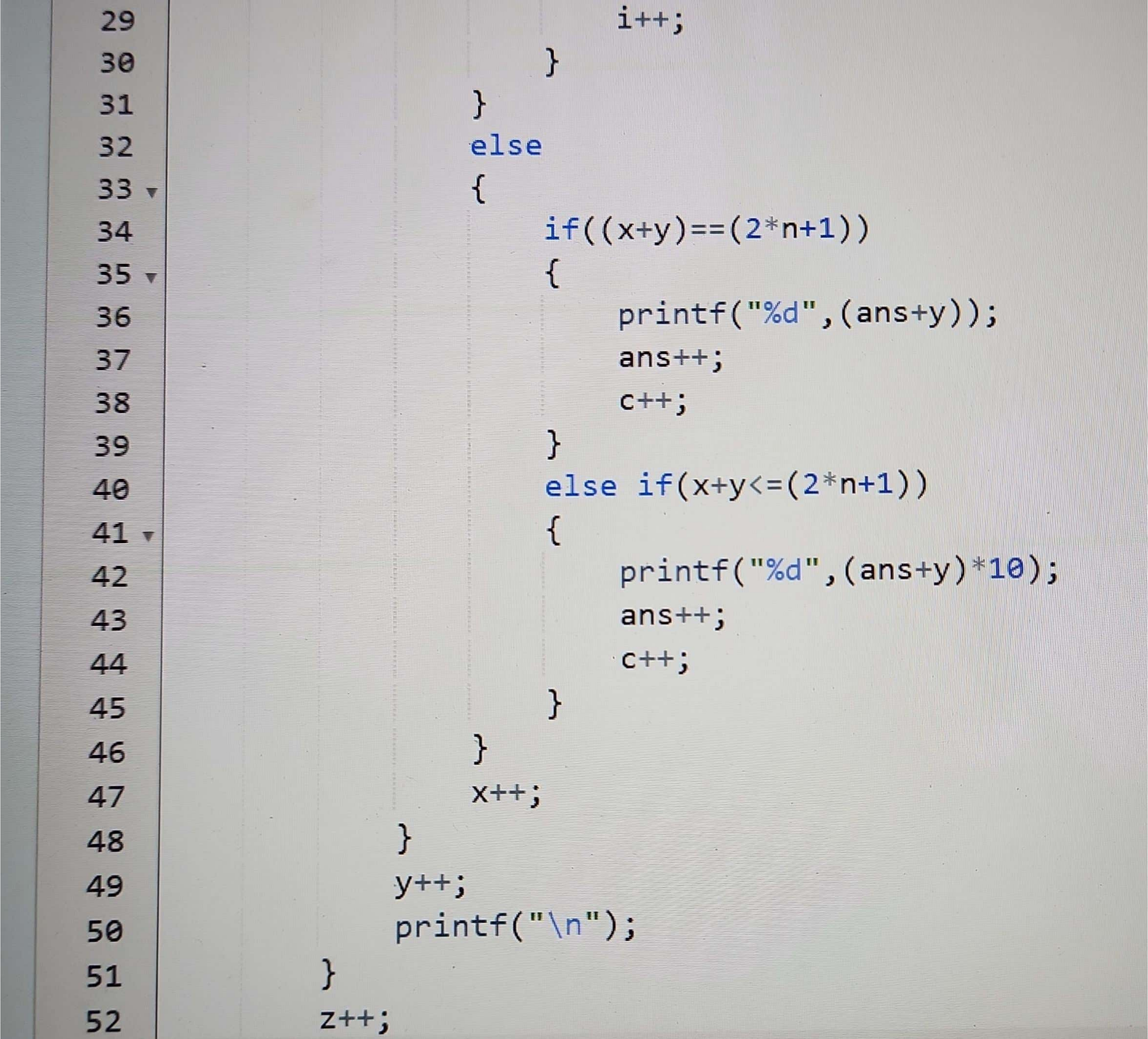


```
10203010011012
**4050809
****607
Case #2
1020304017018019020
**50607014015016
****809012013
******10011
Case #3
102030405026027028029030
**6070809022023024025
****10011012019020021
******13014017018
********15016
```

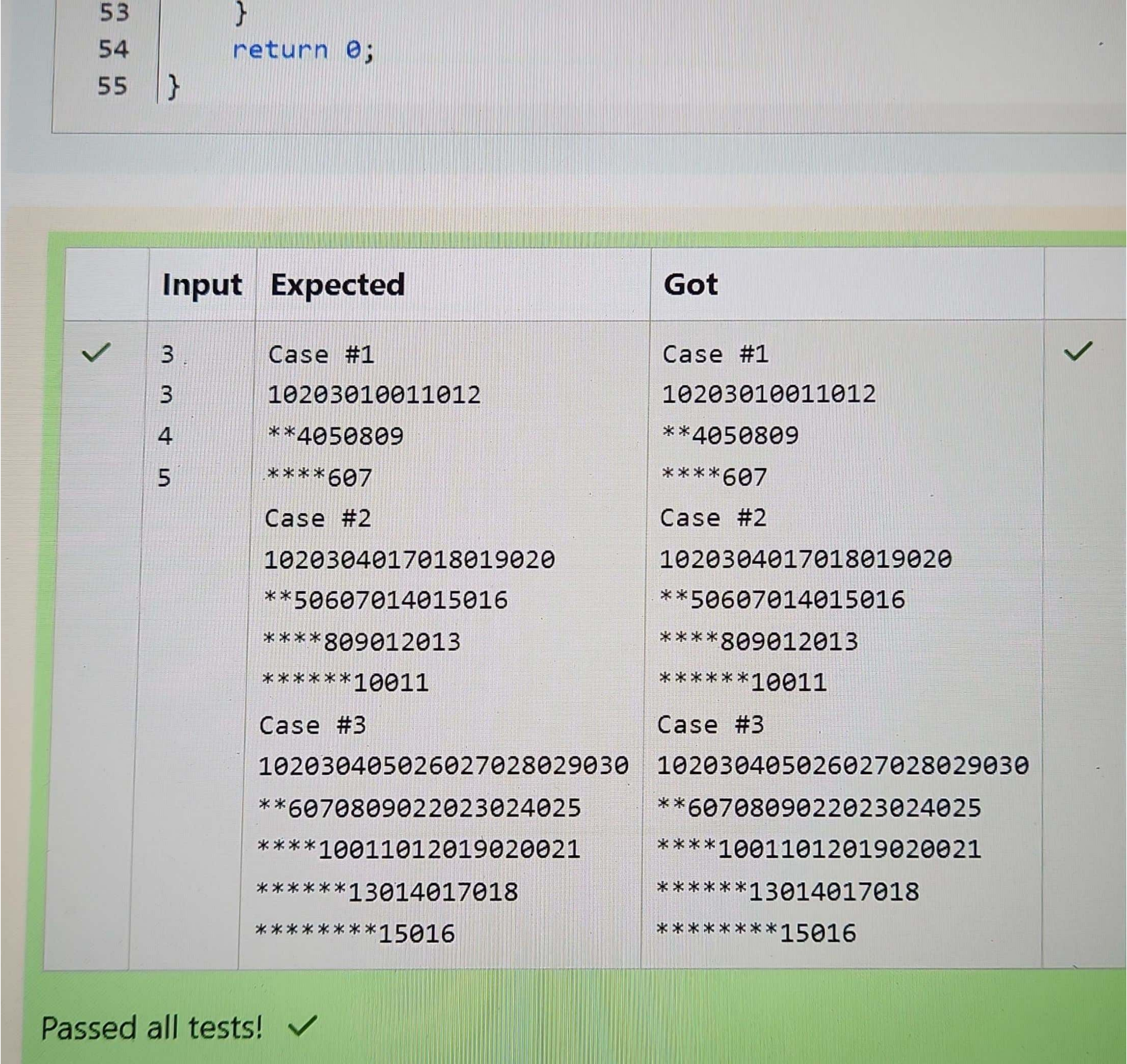
Answer: (penalty regime: 0 %)



```c
#include<stdio.h>
int main()
{
    int t,n,x,y,z=1,i,ans,c;
    scanf("%d",&t);
    while(z<=t)
    {
        scanf("%d",&n);
        printf("Case #%d\n",z);
        y=1;
        i=1;
        c=0;
        while(y<=n)
        {
            x=1;
            ans=(n*n);
            ans=ans-c;
            while(x<=2*n)
            {
                if(x<=n)
                {
                    if(x<y)
                    {
                    printf("**");
                    }
                    else if(x<=n)
                    {
                        printf("%d",i*10);
```



```
                        i++;
                    }
                }
                else
                {
                    if((x+y)==(2*n+1))
                    {
                        printf("%d",(ans+y));
                        ans++;
                        c++;
                    }
                    else if(x+y<=(2*n+1))
                    {
                        printf("%d",(ans+y)*10);
                        ans++;
                        c++;
                    }
                }
                x++;
            }
            y++;
            printf("\n");
        }
        z++;
```



```
    }
    return 0;
}
```

	Input	Expected	Got	
✓	3 3 4 5	Case #1 10203010011012 **4050809 ****607 Case #2 1020304017018019020 **50607014015016 ****809012013 ******10011 Case #3 1020304050260270280029030 **6070809022023024025 ****10011012019020021 ******13014017018 ********15016	Case #1 10203010011012 **4050809 ****607 Case #2 1020304017018019020 **50607014015016 ****809012013 ******10011 Case #3 1020304050260270280029030 **6070809022023024025 ****10011012019020021 ******13014017018 ********15016	✓

Passed all tests! ✓