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,d,i=0,i1,i2,n;
    char c;
    scanf("%d",&T);
    while(i<T)
    {
        scanf("%d",&d);
        i1=0;
        while(i1<d)
        {
            n=1;
            i2=0;
            if(i1%2==0)
            {
                n=0;
            }
            while(i2<d)
            {
                c='B';
                if(i2%2==n)
                {
                    c='W';

                }
            printf("%c",c);
            i2++;
        }
        i1+=1;
        printf("\n");
        }
    i=i+1;
    }
}
```

	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! ✓

Question **2**
Correct
Marked out of 5.00

Let's print a chessboard!

Write a program that takes input:

5	WBW WBWBW BWBWB WBWBW BWBWB WBWBW	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,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++)
        {
            z=(s=='W')?0:1;
            o=(i1%2==z)?0:1;
            for(i2=0;i2<d;i2++)
            {
                c=(i2%2==o)?'W':'B';
                printf("%c",c);
            }
            printf("\n");
        }
    }
    return 0;
}
```

	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

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 num,t;
    scanf("%d",&t);
    int st1=1;
    int st2;
    for(int k=1;k<=t;k++)
    {
        printf("Case #%d\n",k);
        scanf("%d",&num);
        st1=1;
        st2=num*(num+1);
        for(int i=0;i<num;i++)
        {
            for(int j=0;j<i;j++){
                printf("**");
            }
            for(int j=0;j<num-i;j++){
                printf("%d",(st1++)*10);
            }
            st2=st2-(num-i-1);
            for(int j=0;j<(num-i-1);j++){
                printf("%d",(st2++)*10);
            }
            printf("%d",st2);
            st2=st2-(num-i);
            printf("\n");
        }
    }
    return 0;
}
```

	Input	Expected	Got
✓	3 3 4 5	Case #1 10203010011012 **4050809 ****607 Case #2 1020304017018019020 **50607014015016 ****809012013 ******10011 Case #3 102030405026027028029030 **6070809022023024025 ****10011012019020021 ******13014017018 ********15016	Case #1 102030100110 **4050809 ****607 Case #2 102030401701 **5060701401 ****80901201 ******10011 Case #3 102030405026 **6070809022 ****10011012 ******130140 ********1501

Passed all tests! ✓

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

Note:

1 <= N <= 10^8

Answer: (penalty regime: 0 %)

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

	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 %)

```c
#include<stdio.h>
int main(){
    int  rn,n,nt=0,i=0;
    scanf("%d",&n);
    do{
        nt=n;rn=0;
```

	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 %)

```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! ✓

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:

33

Explanation:

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

Sample Input 2:

34

Sample Output 2:

	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:

33

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 %)

```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);
    return 0;
}
```

	Input	Expected	Got	
✓	34	33344	33344	✓

Passed all tests! ✓