

WBW
BWB
WBW
WBWBW
BWBWB
WBWBW
BWBWB
WBWBW

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2 int main()
3 {
4     int T, d, i = 0, l1, l2, o;
5     char c;
6     scanf("%d", &T);
7     while(l1 < T)
8     {
9         scanf("%d", &d);
10        l1 = 0;
11        while(l1 < d)
12        {
13            o = 1;
14            l2 = 0;
15            if(l1 % 2 == 0)
16            {
17                o = 0;
18            }
19            while(l2 < d)
20            {
21                c = 'B';
22                if(l2 % 2 == o)
23                {
24                    c = 'W';
25                }
26                printf("%c", c);
27                l2++;
28            }
29            l1++;
30            printf("\n");
31        }
32        l = l + 1;
33    }
34    return 0;
35 }
```

	Input	Expected	Got	
✓	2	WBW	WBW	✓
	3	BWB	BWB	
	5	WBWBW	WBWBW	
		WBWBWBW	WBWBWBW	
		WBWBWBWBW	WBWBWBWBW	
		WBWBWBWBWBW	WBWBWBWBWBW	
		WBWBWBWBWBWBW	WBWBWBWBWBWBW	

Passed all tests! ✓

Output Format

Print the chessboard as per the given examples

Sample input / Output

Input:

```
2
2 W
3 B
```

Output:

```
WB
BW
BWB
WSW
BWB
```

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int T,0,1,11,12,0,x;
5     char c,s;
6     scanf("%d",&T);
7     for(i=0;i<T;i++)
8     {
9         scanf("%d %c",&d,&c);
10        for(i1=0;i1<d;i1++)
11        {
12            x=(x+'W')%0;1;
13            o=(1112--x)%0;1;
14            for(i2=0;i2<d;i2++)
15            {
16                c=(1212--o)%2?"W":"B";
17                printf("%c",c);
18            }
19            printf("\n");
20        }
21    }
22    return 0;
23 }
```

Case #2

102030405026027028029030

**0070809022023024025

****10011012019020021

*****13014017018

*****15016

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2 int main()
3 {
4     int n,v,p3,c,ln,l,li,l2,t,ti;
5     scanf("%d",&t);
6     for(ti=0;ti<t;ti++){
7         v=0;
8         scanf("%d",&n);
9         printf("Case #li\n",ti+1);
10        for(i=0;i<n;i++){
11            c=0;
12            if(i%4){
13                for(li=0;li<i;li++) printf(" ");
14            }
15            for(l1=1;l1<n;l1++){
16                if(l1%4) c++;
17                printf("%d",c+v);
18            }
19            if(l1%4){
20                p3=v+(v*(v-1))+1;
21                ln=p3;
22            }
23            ln=ln-c;
24            p3=ln;
25            for(l2=1;l2<n;l2++){
26                printf("%d",p3++);
27                if(l2%4==3) printf(" ");
28            }
29            printf("\n");
30        }
31    }
32    return 0;
33 }
```

	Input	Expected	Got	
✓	3	Case #1	Case #1	✓
	3	102030405026027028029030	102030405026027028029030	
	4	**0070809022023024025	**0070809022023024025	
	5	****10011012019020021	****10011012019020021	
		*****13014017018	*****13014017018	
		*****15016	*****15016	

Passed all tests! ✓

Output:

false

Explanation:

123 is a 3-digit number, and $123 \leq 1^2 \cdot 3 + 2^2 \cdot 3 + 3^2 \cdot 3 = 36$.

Example 3:

Input:

1634

Output:

true

Note:

 $1 \leq N \leq 10^8$

Answer: (penalty regime: 0 %)

```

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

```

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

Passed all tests! ✓

Take a number, reverse it and add it to the original number until the obtained number is a palindrome. Constraints $1 \leq \text{num} \leq 99999999$ Sample Input 1 32 Sample Output 1 55 Sample Input 2 789 Sample Output 2 66066

Answer: (penalty regime: 0 %)

```

1 #include<stdio.h>
2 int main()
3 {
4     int rn,n,nt=0,i=0;
5     scanf("%d",&n);
6     do{
7         nt=n;rn=0;
8         while(n!=0){
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! ✓

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.

Sample Output 1:

33

Explanation:

Here the lucky numbers are 2, 4, 23, 34, and the 3rd lucky number is 33.

Sample input 2:

34

Sample Output 2:

33344

Answer: (penalty regime: 0 %)

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