

Status	Finished
Started	Monday, 3 November 2025, 7:02 PM
Completed	Monday, 3 November 2025, 7:19 PM
Duration	17 mins 25 secs

Question **1**

Correct

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 %)

```
1  #include<stdio.h>
2  int main(){
3  int n,size,i,j,count;
4  scanf("%d",&n);
5  while(n--){
6  scanf("%d",&size);
7  count=0;
8  for(i=0;i<size;i++){
9  for(j=0;j<size;j++){
10 if(++count%2==1)
11 printf("W");
12 }
```

```
12 else
13 printf("B");
14 }
15 if(size%2==0)
16 count++;
17 printf("\n");
18 }
19 }
20 }
21
```



	Input	Expected	Got	
✓	2	WBW	WBW	✓
	3	BWB	BWB	
	5	WBW	WBW	
		WBWBW	WBWBW	
		BWBWB	BWBWB	
		WBWBW	WBWBW	
		BWBWB	BWBWB	
		WBWBW	WBWBW	



Passed all tests! ✓

Question **2**

Correct

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 %)

```
1  #include<stdio.h>
2  int main(){
3      int t;
4      scanf("%d",&t);
5      while(t--){
6          int n;
7          char ch;
8          scanf("%d %c",&n,&ch);
9          for(int i=0;i<n;i++){
10             for(int j=0;j<n;j++){
11                 if((i+j)%2 == 0)
12                     printf("%c",ch);
13                 else
```

```
14         printf("%c", (ch == 'W') ? 'B' : 'W');  
15     }  
16     printf("\n");  
17 }  
18 }  
19 return 0;  
20 }
```

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

Passed all tests! ✓

Question **3**

Correct

Problem Statement:

In a small coding competition, participants are to be grouped into teams of three members, each member represented by a number — 1, 2, and 3.

The rule of the competition states that no member can repeat within the same team.

Write a program to display all possible unique team combinations that can be formed using the members 1, 2, and 3 without repetition.

Sample Output:

1 2 3

1 3 2

2 1 3

2 3 1

3 1 2

3 2 1

Answer: (penalty regime: 0 %)

```
1  #include<stdio.h>
2  int main() {
3      printf("1 2 3\n");
4      printf("1 3 2\n");
5      printf("2 1 3\n");
6      printf("2 3 1\n");
7      printf("3 1 2\n");
8      printf("3 2 1\n");
9      return 0;
10 }
```



	Expected	Got	
✓	1 2 3 1 3 2 2 1 3 2 3 1 3 1 2 3 2 1	1 2 3 1 3 2 2 1 3 2 3 1 3 1 2 3 2 1	✓

Passed all tests! ✓