

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 different values for size of the chessboard

Output format:

Print a chessboard of dimensions size * size. 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;
4     scanf("%d", &n);
5     for (int k=0; k<n; k++) {
6         scanf("%d", &size);
7         for (int i=0; i<size; i++) {
8             for (int j=0; j<size; j++) {
9                 if ((i+j)%2==0) {
10                     printf("W");
11                 } else {
12                     printf("B");
13                 }
14             }
15             printf("\n");
}
```

```
16 }  
17 }  
18 }
```



	Input	Expected	Got	
✓	2	WBW	WBW	✓
	3	BWB	BWB	
	5	WBW WBWBW BWBWB WBWBW BWBWB WBWBW	WBW WBWBW BWBWB WBWBW BWBWB 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 n;
4     scanf("%d", &n);
5     for (int k=0; k<n; k++) {
6         int size;
7         char start;
8         scanf("%d %c", &size, &start);
9         for (int i=0; i<size; i++) {
10            for (int j=0; j<size; j++) {
11                if ((i+j)%2==0) {
12                    printf("%c", start);
13                } else {
14                    printf("%c", (start=='W'? 'B': 'W'));
15                }
16            }
17        }
18    }
19 }
```

```
17 }  
18 }  
19 }  
20 return 0;  
21 }
```

[]

	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
4     int i,j,k;
5     for (i=1;i<=3;i++) {
6         for (j=1;j<=3;j++) {
7             for (k=1;k<=3;k++) {
8                 if (i!=j && j!=k && i!=k) {
9                     printf("%d %d %d\n", i, j, k);
10                }
11            }
12        }
13    }
14
15    return 0;
16 }
```



	Expected	Got	
1	1 2 3	1 2 3	1
2	1 3 2	1 3 2	2
3	2 1 3	2 1 3	3
4	2 3 1	2 3 1	4
5	3 1 2	3 1 2	5
6	3 2 1	3 2 1	6

Passed all tests! 1