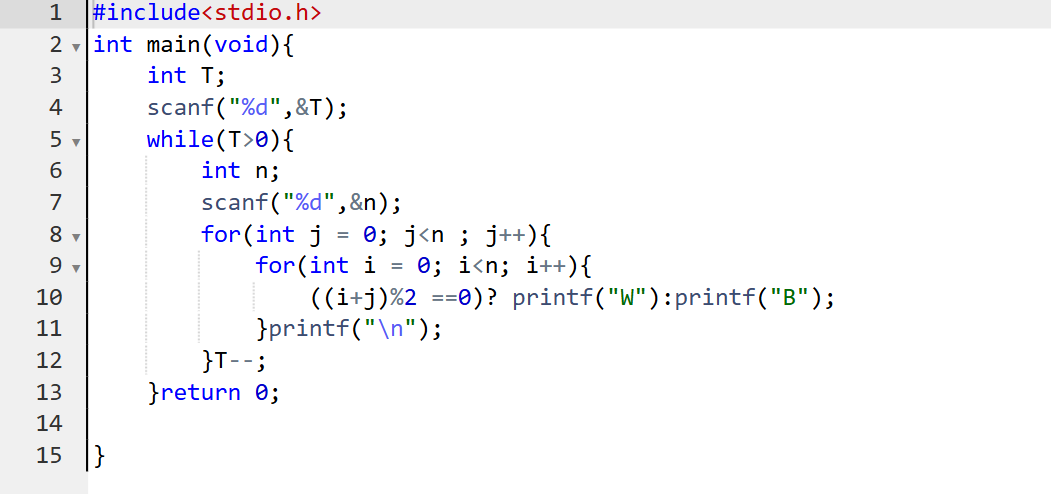
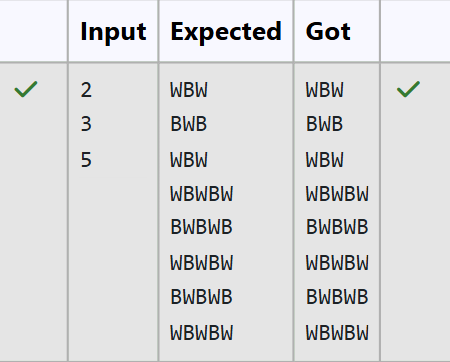
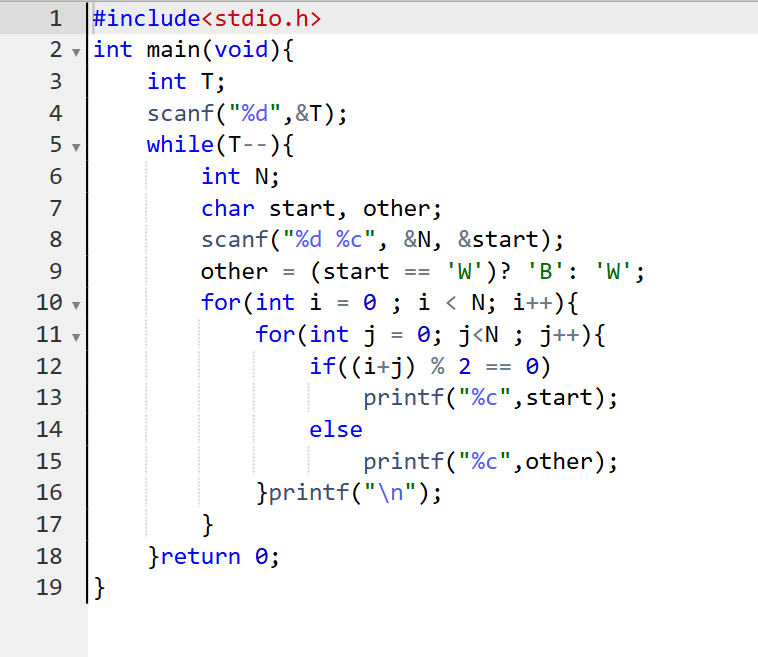
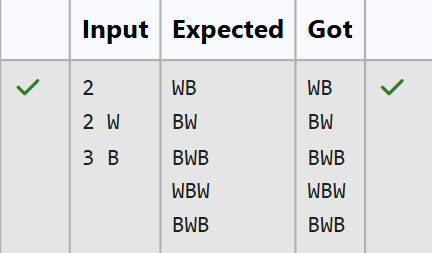
|  |
| --- |
| **Simple Chessboard** |
| **Problem Statement:**  Write a program that prints a simple chessboard.  Input format:  The first line contains the number of inputs T.  The lines after that contain a different value for size of the chessboard  Output format:  Print a chessboard of dimensions size \* size.  Print W for white spaces and B for black spaces.  Sample Input:  2  3  5  Sample Output:  WBW  BWB  WBW  WBWBW  BWBWB  WBWBW  BWBWB  WBWBW |



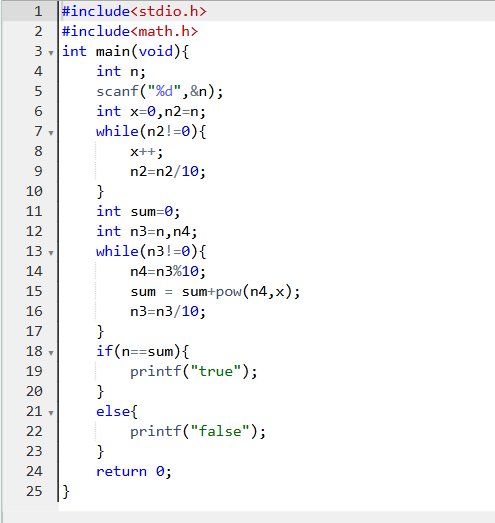


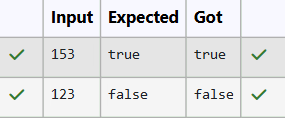
|  |
| --- |
| **Print Our Own Chessboard** |
| **Problem Statement:**  Let’s print a chessboard!  Write a program that takes input:  The first line contains T, the number of test cases  Each test case contains an integer N and also the starting character of the chessboard  Output Format  Print the chessboard as per the given examples  Sample Input:  2  2 W  3 B  Sample Output:  WB  BW  BWB  WBW  BWB |





|  |  |
| --- | --- |
|  | **Pattern Printing** |
| **Problem Statement:**  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:  1020304017018019020  \*\*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 Format  First line print Case #i where i is the test case number, In the subsequent line, print the pattern  Sample Input 3  3  4  5  Sample Output Case #1  10203010011012  \*\*4050809  \*\*\*\*607  Case #2  1020304017018019020  \*\*50607014015016  \*\*\*\*809012013  \*\*\*\*\*\*10011  Case #3  102030405026027028029030  \*\*6070809022023024025  \*\*\*\*10011012019020021  \*\*\*\*\*\*13014017018  \*\*\*\*\*\*\*\*15016 | |
|  | **Armstrong Number** |
| **Problem Statement:**  The k-digit number N is an Armstrong number if and only if the k-th power of each digit sums to N.  Given a positive integer N, return true if and only if it is an Armstrong number.  **Note**: 1 <= N <= 10^8  **Hint**: 153 is a 3-digit number, and 153 = 1^3 + 5^3 + 3^3. Sample Input:  153  Sample Output: true  Sample **Input**: 123  Sample **Output**: false  Sample **Input**: 1634  Sample **Output**: true | |





|  |
| --- |
| **Reverse and Add Until Get a Palindrome** |
| **Problem Statement:**  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 |

|  |  |
| --- | --- |
|  | **Lucky Number** |
| **Problem Statement:**  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 | |

