WEEK 05

1. 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.

Program:

KARPAGASUNDARI V 2024-CSBS

Microsoft Edge

/review.php?attempt=119811&cmid=122

```c
#include<stdio.h>
int main(){
    int t;
    scanf("%d", &t);
    for(int k = 0; k<t; k++){
        int n;
        char p;
        scanf("%d", &n);
        for(int row = 0; row <n; row++){
            p = (row%2 == 0)? 'W':'B';
            for(int col =0; col<n; col++){
                printf("%c", p);
                p = (p == 'W')? 'B':'W';
            }
        printf("\n");
    }
   }
   return 0;
}
```

Output:

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

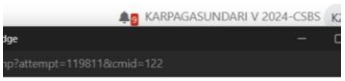
Passed all tests! ✓

2. 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

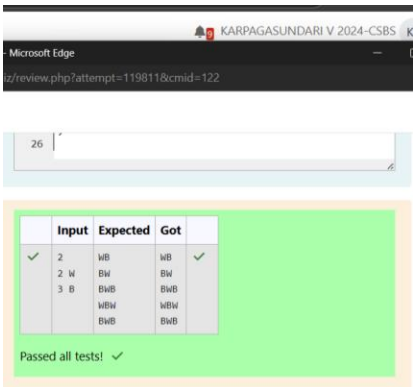
Output Format : Print the chessboard as per the given examples

Program:




```c
#include<stdio.h>
int main(){
    int t;
    scanf("%d", &t);
    for(int k = 0; k<t; k++){
        int n;
        char p;
        scanf("%d", &n);
        for(int row = 0; row <n; row++){
            p = (row%2 == 0)? 'W':'B';
            for(int col =0; col<n; col++){
                printf("%c", p);
                p = (p == 'W')? 'B':'W';
            }
        printf("\n");
    }
  }
  return 0;
}
```

Output:

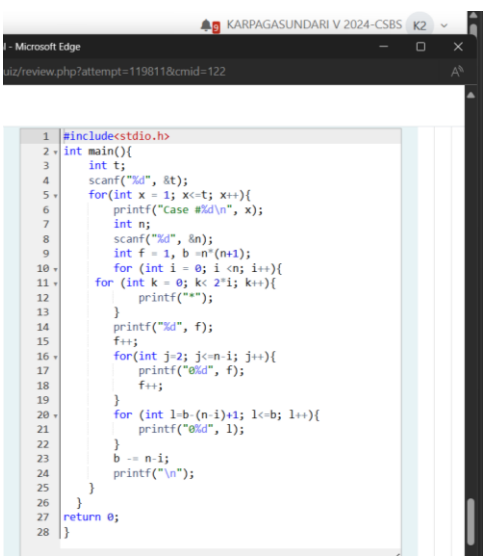


3. Decode the logic and print the Pattern that corresponds to given input.

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

Program:



Output:

	Input	Expected	Got
✓	3 3 4 5	Case #1 10203010011012 **4050809 ****607 Case #2 1020304017018019020 **50607014015016 ****809012013 ******10011 Case #3 102030405026027028029030 **607080902202302402 5 ****10011012019020021 ******13014017018 ********15016	Case #1 10203010011012 **4050809 ****607 Case #2 1020304017018019020 **50607014015016 ****809012013 ******10011 Case #3 102030405026027028029030 **607080902202302402 5 ****1001101201902002 1 ******13014017018 ********15016

Passed all tests! ✓

4. 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.

Sample Input:

153

Sample Output:

True

Program:

KARPAGASUNDARI V 2024-CSBS

onal - Microsoft Edge

/quiz/review.php?attempt=1199088&cmid=123

```c
#include<stdio.h>
#include<math.h>
int main(){
    int n,k = 0, sum = 0;
    scanf("%d", &n);
    int temp = n;
    while (temp >0){
        k++;
        temp /= 10;
    }
    temp = n;
    while (temp>0){
        sum+= pow(temp%10,k);
        temp /= 10;
    }
    if(sum == n){
        printf("true");
    }
    else{
        printf("false");
    }
    return 0;
}
```

Output:

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

Passed all tests! ✓

5. Take a number, reverse it and add it to the original number until the obtained number is a palindrome.

Program:

KARPAGASUNDARI V 2024-CSBS

Microsoft Edge

review.php?attempt=1199088&cmid=123

```c
#include<stdio.h>
long long revnum(long long num){
    long long rev = 0;
    while (num>0){
        rev = rev*10 + (num%10);
        num /= 10;
    }
    return rev;
}
int ispalindrome(long long num){
    return num == revnum(num);
}
int main(){
    long long num;
    scanf("%lld", &num);
    while (1){
      long long rev = revnum(num);
      num = num+rev;
      if(ispalindrome(num)){
          printf("%lld", num);
          break;
      }
    }
return 0;
}
```

Output:

	Input	Expected	Got	
✓	32	55	55	✓
✓	789	66066	66066	✓

Passed all tests! ✓

6. 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.

Program:

KARPAGASUNDARI V 2024-CSBS

soft Edge

ew.php?attempt=119908&cmid=123

```c
#include<stdio.h>
#include<math.h>
int main(){
    int n, k=1, num=0;
    scanf("%d", &n);
    while(n>0){
        if (n-pow(2,k) >= 0){
            n = n- pow(2,k);
            k++;
        }
        else{
            break;
        }
    }
    int t = 0, p=3;
    for (int i =0; i<k; i++){
        t=0;
        p=3;
        for (int j=1; j<=n; j++){
            if (t == pow(2,i)){
                t=1;
                p= (p==3) ? 4:3;
            }
            else{
                t++;
            }
        }
        num = p*pow(10,i) + num;
    }
    printf("%d", num);
return 0;
}
```

Output:

	Input	Expected	Got	
✓	34	33344	33344	✓

Passed all tests! ✓