Question **1**

Correct

Marked out of 3.00

Flag question

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 %)

```c
#include<stdio.h>
int  main(){
    int T;
    scanf("%d",&T);
    while(T--)
    {
        int n;
        char ch='W';
        scanf("%d",&n);
        for(int i=1;i<=n;i++){
            for(int j=1;j<=n;j++){
                printf("%c",ch);
                if(ch=='W')
                    ch='B';
                else
                    ch='W';}
                if(n%2==0){
                    if(ch=='W')
                    ch='B';
                    else
                    ch='W';
                }
                printf("\n");
            }
        }
    }
```

	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! ✓

Question **2**

Correct

Marked out of 5.00

Flag question

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 %)

```c
#include<stdio.h>
int main(){
    int T,d,i,i1,i2,o,z;
    char c,s;
    scanf("%d",&T);
    for(i=0;i<T;i++)
    {
        scanf("%d %c",&d,&s);
        for(i1=0;i1<d;i1++){
            z=(s=='W') ? 0:1;
            o=(i1%2==z) ? 0:1;
            for(i2=0;i2<d;i2++){
                c=(i2%2==o) ?'W':'B';
            printf("%c",c);
        }
        printf("\n");
        }
    }
    return 0;
}
```

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

Passed all tests! ✓

Question 3

Correct

Marked out of 7.00

Flag question

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

First line print Case #i where i is the test case number
In the subsequent line, print the pattern

Test Case 1

3
3
4
5

Output

Case #1
10203010011012
**4050809
****607
Case #2
1020304017018019020

Output

Case #1

10203010011012

**4050809

****607

Case #2

1020304017018019020

**50607014015016

****809012013

******10011

Case #3

102030405026027028029030

**6070809022023024025

****10011012019020021

******13014017018

********15016

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int n,v,p3,c,in,i,i1,i2,t,ti;
    scanf("%d",&t);
    for(ti=0;ti<t;ti++){
        v=0;
        scanf("%d",&n);
        printf("Case #%d\n",ti+1);
        for(i=0;i<n;i++){
            c=0;
            if(i>0){
                for(i1=0;i1<i;i1++)
                printf("**");
            }
            for(i1=i;i1<n;i1++){
               if(i>0) c++;
               printf("%d0",++v);
            }
            if(i==0){
                p3=v+(v*(v-1))+1;
                in=p3;
            }
            in=in-c;
            p3=in;
            for(i2=i;i2<n;i2++){
                printf("%d",p3++);
                if(i2!=n-1)
                printf("0");
            }
            printf("\n");
        }
    }
}
```

	Input	Expected	Got	
✓	3 3 4 5	Case #1 10203010011012 **4050809 ****607 Case #2 102030401701801902O **50607014015016 ****809012013 ******10011 Case #3 10203040502602702802903O **6070809022023024025 ****10011012019020021 ******13014017018 ********15016	Case #1 10203010011012 **4050809 ****607 Case #2 102030401701801902O **50607014015016 ****809012013 ******10011 Case #3 10203040502602702802903O **6070809022023024025 ****10011012019020021 ******13014017018 ********15016	✓

Passed all tests! ✓

Question 1

Correct

Marked out of 3.00

Flag question

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.

Example 1:

Input:

153

Output:

true

Explanation:

153 is a 3-digit number, and 153 = 1^3 + 5^3 + 3^3.

Example 2:

Input:

123

Output:

false

Explanation:

123 is a 3-digit number, and 123 != 1^3 + 2^3 + 3^3 = 36.

Example 3:

Input:

1634

Output:

true

Note:

1 <= N <= 10^8

Answer: (penalty regime: 0 %)

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int a,b,c=0,d,i,e=1;
    scanf("%d",&a);
    for(b=a;b>0;b=b/10){
        c++;
    }
    b=a;
    for(d=0;b>0;b=b/10){
        e=1;
        for(i=0;i<c;i++){
            e=e*(b%10);
        }
        d=d+e;
    }
    if(d==a)
    printf("true");
    else
    printf("false");
    return 0;
}
```

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

Passed all tests! ✓

Question 2

Correct

Marked out of 5.00

Flag question

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

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    long long int n,s,r,t,ts;
    scanf("%lld",&n);
    while(1){
        r=0;
        t=n;
        while(n){
            r=r*10+(n%10);
            n=n/10;
        }
        s=t+r;
        ts=s;
        r=0;
        while(s){
            r=r*10+(s%10);
            s=s/10;
        }
        if(ts==r)
        break;
        n=ts;
    }
    printf("%lld",ts);
    return 0;
}
```

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

Passed all tests! ✓

Question 3

Correct

Marked out of 7.00

Flag question

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 n

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

Explanation:

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

Sample Input 2:

34

Sample Output 2:

33344

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int a,b=1,ln=0,i=1,j;
    scanf("%d",&a);
    for(;ln<=a;i++){
        for(j=i;j>0;j=j/10){
            if(j%10==3||j%10==4)
            b=1;
            else{
                b=0;
                break;
            }
        }
        if(b==1){
            ln++;
            if(ln==a){
                printf("%d",i);
                break;
            }
        }
    }
}
```

	Input	Expected	Got	
✓	34	33344	33344	✓

Passed all tests! ✓