NINETHA NATARAJAN N

ECE-D

240801221

Problem Statement 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.

Sample Input:

2

3

5

Sample Output:

WBW

BWB

WBW

WBWBW

BWBWB

WBWBW

BWBWB

WBWBW

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int T,size;
    scanf("%d",&T);
    while(T--){
        scanf("%d",&size);
        for(int i=0;i<size;i++){
            for(int j=0;j<size;j++){
                if((i+j)%2==0){
                    printf("W");
                }
                else{
                    printf("B");
                }
            }
            printf("\n");
        }
    }
    if(T>0){
        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! ✓

Problem Statement 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

Sample Input:

2

2 W

3 B

Sample Output:

WB

BW

BWB

WBW

BWB

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int main(){
    char s;
    int t,n;
    scanf("%d",&t);
    while(t--){
        scanf("%d %c",&n,&s);
        for(int i=0;i<n;i++){
            for(int j=0;j<n;j++){
                if((i+j)%2==0){
                    printf("%c",s);
                }
                else{
                    if(s=='W'){
                        printf("B");
                    }
                    else{
                        printf("W");
                    }
                }
            }
            printf("\n");
        }
    }
}
```

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

Passed all tests! ✓

Activate Windows

Problem Statement 3:

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:

10203040170180190 20

****506070140150 16**

******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

10203040502602702802903 0

****6070809022023024025**

*****100110120190200 21**

*******13014017018**

*********15016**

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int t;
    scanf("%d",&t);
    for(int x = 1;x <= t;x++){
        printf("Case #%d\n",x);
        int n;
        scanf("%d",&n);
        int f=1,b=n*(n+1);
        for(int i=0;i<n;i++){
            for(int k=0;k< 2*i ;k++){
                printf("*");
            }

            printf("%d",f);
            f++;
            for(int j=2;j<= n-i ;j++){
                printf("0%d",f);
                f++;
            }
            for(int l=b-(n-i)+1;l<=b;l++){
                printf("0%d",l);
            }
            b-=n-i;
            printf("\n");
        }
    }
    return 0;
}
```

	Input	Expected	Got	
✓	3 3 4 5	Case #1 10203010011012 **4050809 ****607 Case #2 1020304017018019020 **50607014015016 ****809012013 ******10011 Case #3 10203040502602702802903 0 **6070809022023024025 ****10011012019020021 ******13014017018 ********15016	Case #1 10203010011012 **4050809 ****607 Case #2 1020304017018019020 **50607014015016 ****809012013 ******10011 Case #3 10203040502602702802903 0 **6070809022023024025 ****10011012019020021 ******13014017018 ********15016	✓

Passed all tests! ✓

Problem Statement 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.

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

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
#include <math.h>

int main(){
    int n,sum=0,t,c=0;
    scanf("%d",&n);
    t=n;
    while(n>0){
        n=n/10;
        c=c+1;
    }
    n=t;
    while(t>0){
        sum=sum+pow((t%10),c);
        t/=10;
    }
    if(sum==n){
        printf("true");
    }
    else{
        printf("false");
    }
}
```

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

Passed all tests! ✓

Activate Windows
Go to Settings to activate Window

Problem Statement 5:

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>
#include<limits.h>
long long reversenumber(long long num){
    long long reversed=0;
    while(num>0){
        int digit=num%10;
        if(reversed>(LONG_MAX-digit)/10){
            return -1;
        }
        reversed=reversed*10+digit;
        num/=10;
    }
    return reversed;
}
int ispallindrome(long long num){
    return num==reversenumber(num);
}

int main(){
    long long num;
    scanf("%lld",&num);
    while(1){
    long long reversed=reversenumber(num);
    num=num+reversed;
        if(ispallindrome(num)){
            printf("%lld",num);
            break;

        }
    }
}
```

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

Passed all tests! ✓

Problem Statement 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.

Sample Input 1:

3

Sample Output 1:

33

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int main(){
    int n=1;
    int i=0,ns,c=0,e;
    scanf("%d",&e);
    while(i<e){
        ns=n;
        while(ns!=0){
            c=0;
            if(ns%10!=3&&ns%10!=4){
                c=1;
                break;
            }
            ns/=10;
        }
        if(c==0){
            i++;
        }
        n++;
    }
    printf("%d",--n);
}
```

	Input	Expected	Got	
✓	34	33344	33344	✓

Passed all tests! ✓

Activate Windows
Go to Settings to activate Window