Your attempts

Attempt 1

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Friday, 22 November 2024, 2:17 PM
Duration	31 days 3 hours

Review

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

Answer: (penalty regime: 0 %)

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

            }
            printf("\n");
        }
    }
    return 0;
}
```

	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! ✓

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

```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'
                printf("%c",c);
            }
            printf("\n");
        }
    }
    return 0;
}
```

```c
1  lude<stdio.h>
2  ain()
3  
4  nt T,d,i,i1,i2,o,z;
5  har c,s;
6  canf("%d",&T);
7  or(i=0;i<T;i++)
8  {
9      scanf("%d %c",&d,&s);
10     for(i1=0;i1<d;i1++){
11         z=(s=='W')?0:1;
12         o=(i1%2==z)?0:1;
13         for(i2=0;i2<d;i2++){
14             c=(i2%2==o)?'W':'B';
15             printf("%c",c);
16         }
17         printf("\n");
18     }
19 }
20 eturn 0;
21 
```

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

Passed all tests! ✓

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

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

10203040170180190 20

**50607014015016

```c
#include<stdio.h>
int main()
{
    int v,c=0;
    scanf("%d",&v);
    while(v!=0)
    {
        c++;
        int a;
        scanf("%d",&a);
        int s1=10,s2=(a*a*10)+
        printf("Case #%d\n",c)
        for(int i=0;i<a;i++)
        {
            for(int j=0;j<i;j+
            {
                printf("**");
            }
            for(int j=0;j<a-i;
            {
                printf("%d",s1
                s1+=10;
            }
            for(int j=0;j<a-i;
            {
                if((j+1)==(a-i
                {
                    printf("%d
                }
                else{
                    printf("%d
                }
                }
                s2-=(a-i)*10;
                s2+=10;
                printf("\n");
            }
            v--;
        }
    }

```

```
1
2
3
4
5
6
7
8
9
10  &a);
11  2=(a*a*10)+10;
12  e #%d\n",c);
13  );i<a;i++)
14
15  j=0;j<i;j++)
16
17  ntf("**");
18
19  j=0;j<a-i;j++)
20
21  ntf("%d",s1);
22  =10;
23
24  j=0;j<a-i;j++)
25
26  (j+1)==(a-i))
27
28  printf("%d",((s2+(j*10)))/10);
29
30  e{
31  printf("%d",(s2+(j*10)));
32
33
34  =(a-i)*10;
35  =10;
36  ntf("\n");
37
38
39
40
41
```

	Input	Expected	Go
✓	3 3 4 5	Case #1 10203010011012 **4050809 ****607 Case #2 10203040170180190 20 **506070140150 16 ****809012013 ******10011 Case #3 10203040502602702802 9030 **607080902202302 4025 ****100110120190200 21 ******13014017018 ********15016	Ca 10 ** ** Ca 10 ** ** ** Ca 10 ** ** ** **

Passed all tests! ✓

	Got	
020 28029030 24025 20021 8	Case #1 10203010011012 **4050809 ****607 Case #2 10203040170180190 20 **506070140150 16 ****809012013 ******10011 Case #3 10203040502602702802903 0 **607080902202302402 5 ****100110120190200 21 ******13014017018 ********15016	✓

Passed all tests! ✓

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.

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

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

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

Passed all tests! ✓

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

```c
#include<stdio.h>
int main()
{
    int rn,n,nt=0,i=0;
    scanf("%d",&n);
    do{
        nt=n;
        rn=0;
        while(n!=0)
        {
            rn=rn*10+n%10;
            n=n/10;
        }
        n=nt+rn;
        i++;
    }while(rn!=nt || i==1);
    printf("%d",rn);
    return 0;
}
```

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

Passed all tests! ✓

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

Explanation:

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

Sample Input 2:

34

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 cont(int a)
{
    int c=a;
    while(c!=0)
    {
        int d=c%10;
        if(d!=3 && d!=4)
           return 0;
        c/=10;
    }
    return 1;
}
int main()
{
    int a,b=0;
    scanf("%d",&a);
    while(a!=0)
    {
        b++;
        if(cont(b))
        {
            a--;
        }

   }
   printf("%d",b);
   return 0;
}
```

	Input	Expected	Got	
✓	34	33344	33344	✓

Passed all tests! ✓