

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Tuesday, 12 November 2024, 2:36 PM
Duration	41 days 2 hours

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 %)

```
1 #include<stdio.h>
2 int main(){
3     int n;
4     scanf("%d",&n);
5     for(int i=0; i<n; i++){
6         int a;
7         char c1='B',c2='W';
8         scanf("%d",&a);
9         for(int j=0; j<a; j++){
10             if(j%2!=0)
11                 c1='W',c2='B';
12             else
13                 c1='B',c2='W';
14             for(int k=0; k<a; k++){
15                 if(k%2!=0)
16                     printf("%c",c1);
17                 else
18                     printf("%c",c2);
19             }
20             printf("\n");
21         }
22     }
23     return 0;
24 }
```

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

Passed all tests! ✓

Question 2

Correct

Marked out of
5.00Flag
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 %)

```
1 #include<stdio.h>
2 int main(){
3     int n;
4     scanf("%d",&n);
5     for(int i=0; i<n; i++){
6         int a;
7         char b,c1='W',c2='B',temp;
8         scanf("%d %c",&a,&b);
9         for(int j=0; j<a; j++){
10             if((j%2)==0){
11                 if(b!=c1){
12                     c2=c1;
13                     c1=b;}
14             }
15             else{
16                 temp=c1;
17                 c1=c2;
18                 c2=temp;
19             }
20             for(int k=0; k<a; k++){
21                 if(k%2==0)
22                     printf("%c",c1);
23                 else
24                     printf("%c",c2);
25             }
26             printf("\n");
27         }
28     }
29     return 0;
30 }
```

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

**50607014015016

****809012013

*****10011

Case #3

102030405026027028029030

**6070809022023024025

****10011012019020021

*****13014017018

*****15016

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main(){
3     int n;
4     scanf("%d",&n);
5     for(int i=1; i<=n; i++){
6         int a;
7         scanf("%d",&a);
8         int l=1 s=a t=(a*(a+1))-a+1
```

```

10203010011012
**4050809
****607
Case #2
1020304017018019020
**50607014015016
****809012013
*****10011
Case #3
102030405026027028029030
**6070809022023024025
****10011012019020021
*****13014017018
*****15016

```

Answer: (penalty regime: 0 %)

```

1 #include<stdio.h>
2 int main(){
3     int n;
4     scanf("%d",&n);
5     for(int i=1; i<=n; i++){
6         int a;
7         scanf("%d",&a);
8         int l=1,s=a,t=(a*(a+1))-a+1;
9         printf("Case #%d\n",i);
10        for(int j=0; j<a; j++){
11            int k=2*j,t1=t;
12            while(k>0){
13                printf("%c",'*');
14                k-=1;
15            }
16            for(int p=0; p<s; p++){
17                printf("%d",l);
18                l+=1;
19                printf("%d",0);
20            }
21            for(int q=0; q<s; q++){
22                printf("%d",t1);
23                t1+=1;
24                if(q==(s-1)){
25                    break;
26                }
27                printf("%d",0);
28            }
29            s-=1;
30            t-=s;
31            printf("\n");
32        }
33    }
34
35    return 0;
36 }

```

put	Expected	Got
	Case #1	Case #1
	10203010011012	10203010011012
	**4050809	**4050809
	****607	****607
	Case #2	Case #2
	1020304017018019020	1020304017018019020
	**50607014015016	**50607014015016
	****809012013	****809012013
	*****10011	*****10011
	Case #3	Case #3
	102030405026027028029030	1020304050260270280;
	**6070809022023024025	**60708090220230240;
	****10011012019020021	****100110120190200;
	*****13014017018	*****13014017018
	*****15016	*****15016

Passed all testcases

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Saturday, 23 November 2024, 11:12 AM
Duration	30 days 6 hours

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 $123 \neq 1^3 + 2^3 + 3^3 = 36$.

Example 3:

Input:

1634

Output:

true

Note:

$1 \leq N \leq 10^8$

Answer: (penalty regime: 0 %)

```

1 #include<stdio.h>
2 int main(){
3     int n,k=0,a=0;
4     scanf("%d",&n);
5     int n1=n,n2=n;
6     while(n1>0){
7         k+=1;
8         n1/=10;
9     }
10    while(n2>0){
11        int b=n2%10,c=1;
12        for(int i=1;i<=k; i++){
13            c*=b;
14        }
15        a+=c;
16        n2=n2/10;
17    }
18    if(a==n)
19        printf("true\n");
20    else
21        printf("false\n");
22    return 0;
23 }
```

Output:

true

Note:

$1 \leq N \leq 10^8$

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main(){
3     int n,k=0,a=0;
4     scanf("%d",&n);
5     int n1=n,n2=n;
6     while(n1>0){
7         k++;
8         n1/=10;
9     }
10    while(n2>0){
11        int b=n2%10,c=1;
12        for(int i=1;i<=k; i++){
13            c*=b;
14        }
15        a+=c;
16        n2=n2/10;
17    }
18    if(a==n){
19        printf("true");
20    }
21    else{
22        printf("false");
23    }
24    return 0;
25 }
```

	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 $1 \leq \text{num} \leq 99999999$ Sample Input 1 32 Sample Output 1 55 Sample Input 2 789 Sample Output 2 66066

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main(){
3     long long int n,s,rev,temp1,temp2;
4     scanf("%lld",&n);
5     while(1){
6         temp1=n,rev=0;
7         while(n){
8             rev=rev*10+(n%10);
9             n=n/10;
10        }
11        s=temp1+rev;
12        temp2=s,rev=0;
13        while(s){
14            rev=rev*10+(s%10);
15            s=s/10;
16        }
17        if(temp2==rev){
18            break;
19        }
20        n=temp2;
21    }
22    printf("%lld",temp2);
23    return 0;
24 }
```

	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 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

Sample Output 2:

33344

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main(){
3     int n;
4     scanf("%d",&n);
5     int x[n],c=0,g=0,s=0;
6     while(c<n){
7         int r;
8         (c%2==0)?(r=3):(r=4);
9         if((c%2==0)&&(c!=0)){
10             s=x[g];
11             g++;
12         }
13         x[c]=(s*10)+r;
14         c++;
15     }
16     printf("%d",x[n-1]);
17     return 0;
18 }
```

	Input	Expected	Got	
✓	34	33344	33344	✓

Passed all tests! ✓