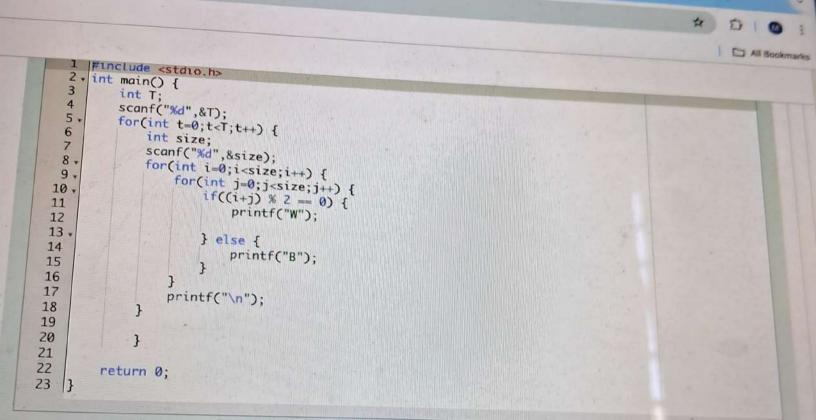


Output format:				
Print a chessboard of dimensions size	e * size. Print a	Print W for whi	te spaces and B for	black spaces.
Input:				
2				
3				
5				
Output:				
WBW				
BWB				
WBW				
WBWBW				
BWBWB				
WBWBW				
BWBWB				

WBWBW



	Input	Expected	Got	
~	2	WBW	WBW.	~
	3	<b>EWB</b>	BWB	
	5	WBW	WBW	
		WBWBW	WBWBW	
		BWBWB	BWBWB	
		WBWBW	WBWBW	
		BWBWB	BWBWB	
		WBWBW	WBWBW	

Passed all testel

Question 2 Let's print a chessboard! Correct Marked out of 5.00 Write a program that takes input: P Flag question 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 W 3 B Output: WB BW

question

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

```
nswer: (penalty regime: 0 %)
     #include<stdio.h>
  2 - int main() {
  3
         int T;
  4
          scanf("%d",&T);
  5
          for(int t=0;t<T;t++) {
  6 7
              int N;
              char start;
  8
              scanf("%d %c",&N,&start);
  9
              char alt=(start == 'W') ? 'B' : 'W';
 10 -
              for(int i=0;i<N;i++) {
 11 +
                   for(int j=0; j<N; j++) {</pre>
  12 .
                       if((i+j)\%2 == 0) {
  13
                            printf("%c", start);
  14 +
                        } else {
  15
                            printf("%c",alt);
  16
  17
  18
                   printf("\n");
  19
  20
  21
           return 0;
  22
```

	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

P 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

```
Test Case 1
3
3
Output
Case #1
 10203010011012
 **4050809
 ****607
 Case #2
 1020304017018019020
 **50607014015016
 ****809012013
 ******10011
 Case #3
 102030405026027028029030
  **6070809022023024025
  ****10011012019020021
  ******13014017018
  *******15016
```

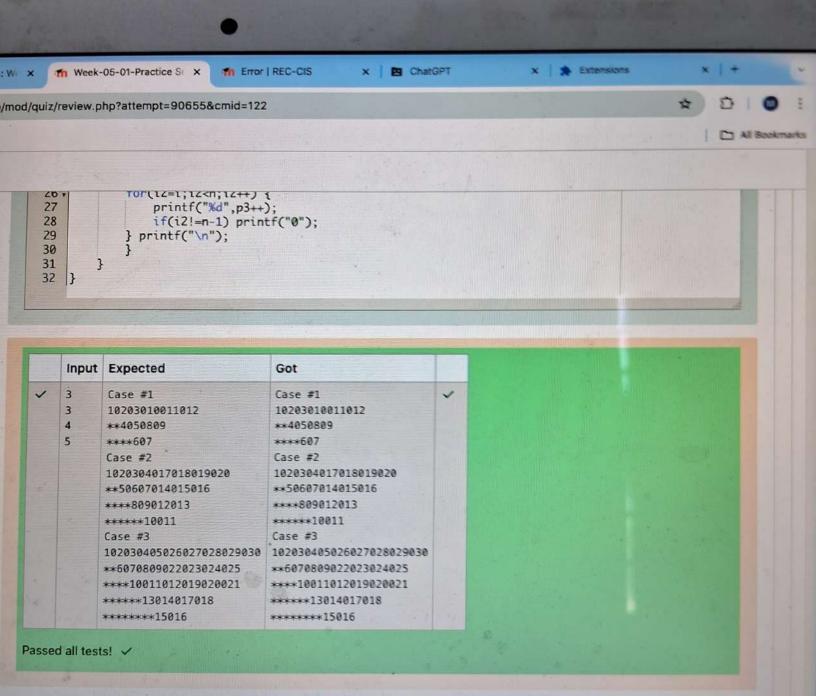
公



☐ All Bookmark

## Answer: (penalty regime: 0 %)

```
#include<stdio.h>
   2 · int main() {
   3
          int n,v,p3,c,in,i,il,i2,t,ti;
   4
          scanf("%d",&t);
   5 .
          for(ti=0;ti<t;ti++) {
   6
              V=0;
   7
               scanf("%d",&n);
   8
              printf("Case #%d\n",ti+1);
  9 .
              for(i=0;i<n;i++) {
 10
                   c=0:
 11 .
                   if(i>0) {
 12
                       for(il=0;il<i;il++) printf("**");
 13
 14
 15
 16 .
              for(il=i;il<n;il++) {</pre>
 17
                  if(i>0) c++;
 18
                  printf("%d0",++v);
19
20 .
              if(i==0) {
21
                  p3=v+(v*(v-1))+1;
22
                  in=p3;
23
24
             in=in-c;
25
             p3=in;
26 .
             for(i2=i;i2<n;i2++) {
                  printf("%d",p3++);
27
28
                  if(i2!=n-1) printf("0");
29
               printf("\n");
30
31
32
```



Question 1 Correct Marked out of 3.00 P 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:

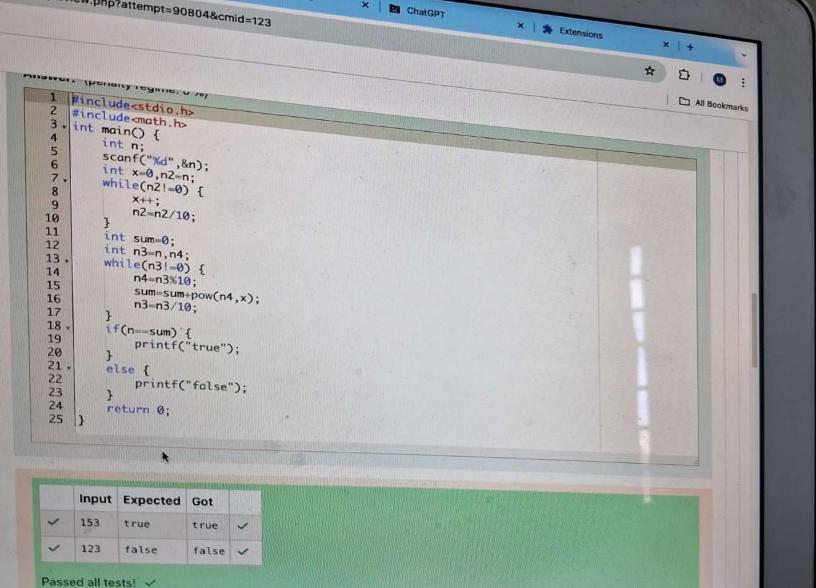
153

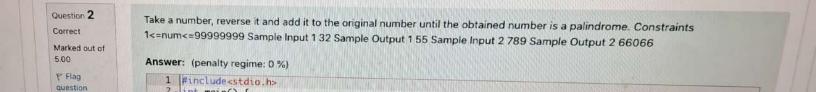
Output:

true

Explanation:

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





```
All Bookmarks
rake a number, reverse it and add it to the original number until the obtained number is a painturbine. Constraints
1<=num<=99999999 Sample Input 1 32 Sample Output 1 55 Sample Input 2 789 Sample Output 2 66066
Answer: (penalty regime: 0 %)
       #include<stdio.h>
       int main() {
            int rn,n,nt=0, i=0;
    4
            scanf("%d", &n);
    5
            do {
    6
                 nt=n; rn=0;
    7.
                 while(n!=0) {
                      rn=rn*10 + n%10;
     9
                      n=n/10;
   10
   11
                 n=nt+rn;
   12
                 i++;
   13
    14
             while(rn!=nt || i==1);
    15
             printf("%d",rn);
    16
             return 0;
    17
         Input Expected Got
         32
                55
                           55
         789
                66066
                           66066
```

123

Passed all tests! ~

Question 3			
orrect	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.		
arked out of	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.		
00	Note that 13, 40 etc., are not lucky as they have other numbers in it.		
Flag			
question	The program should accept a number 'n' as input and display the nth lucky number	as output	
		as output.	
	Sample Input 1:		
	3		
	Sample Output 1:		
	- Campio Galpa		
	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		

