WEEK 5

Q1)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

Q2)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

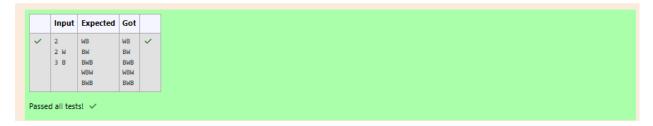
BW

BWB

WBW

BWB

```
Answer: (penalty regime: 0 %)
     1 #include<stdio.h>
      2 - int main(){
                 int a,b,i,j;
                int a,b,1,];
char ch1,ch2;
scanf("%d",&a);
for(i=0;i<a;i++){
    scanf("%d %c",&b,&ch1);
    if(ch1=='W'){
        ch2='B';
    }</pre>
    10
    11 ,
                        else{
                              ch2='W';
    12
    13
                        for(j=0;j<b;j++){
                              for(int k=0;k<b;k++){
    if((k+j)%2==0){
        printf("%c",ch1);
    15 .
    16
    18
    19
                                     else{
                                       printf("%c",ch2);
    20
    21
                              }printf("\n");
    22
23
    25 }
```



Q3)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 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
102030405026027028029030
**6070809022023024025
****10011012019020021
*****13014017018
*****15016
```

```
Answer: (penalty regime: 0 %)
        #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++){
    5
                   v=0;
scanf("%d",&n);
                   printf("Case #%d\n",ti+1);
for(i=0;i<n;i++){</pre>
   10
                        if(i>0){
   11
   12
13
                             for(i1=0;i1<i;i1++) printf("**");</pre>
   14
                   for(i1=i;i1<n;i1++){
                        if(i>0) c++;
printf("%d0",++v);
   15
   16
17
   18
                   if(i==0){
                        p3=V+(V*(V-1))+1;
in=p3;
   19
   20
   21
   22
23
                   in=in-c;
                   p3=in:
   24
                   for(i2=i;i2<n;i2++){
                   printf("%d",p3++);
  if(i2!=n-1) printf("0");
}printf("\n");
   25
   26
27
   28
   29
30
```



Q4)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

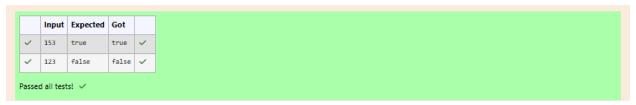
Sample Input:

1634

Sample Output:

true

```
Answer: (penalty regime: 0 %)
        #include<stdio.h>
         #include<math.h>
        int main(){
               int a,n1,x=0,count=0;
scanf("%d",&a);
               n1=a;
while(n1!=0){
                     x++;
n1=n1/10;
   10
11
12 +
              }
int n2=a,n3;
while(n2!=0){
    n3=n2%10;
    count=cour
                     count=count+pow(n3,x);
n2=n2/10;
   14
15
   16
               }
if(a==count){
  printf("true");
   17
18
   19
   20
21
                     printf("false");
   22 23 }
```



Q5)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

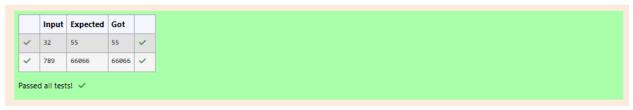
Sample Input 2

789

Sample Output 2

66066

```
Answer: (penalty regime: 0 %)
    1 #include<stdio.h>
    2 - int main(){
3     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;
   10
   11
                  n=nt+rn;
   12
13
                  i++;
   14
15
16
17 }
             while(rn!=nt || i==1);
printf("%d",rn);
return 0;
```



Q6)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

