Name: Santhosh G

Reg No: 240701473

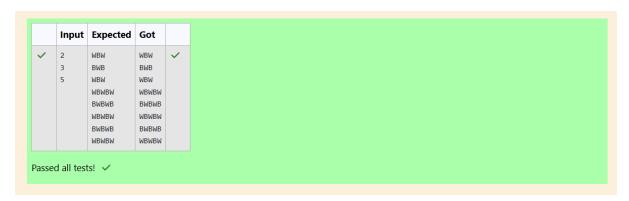
Week 5: Nested Loops - while and for, Jumps in Loops

1. single chessboard

Problem statement:

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 WBWBW BWBWB WBWBW BWBWB WBWBW

```
#include<stdio.h>
int main()
{
 1
2
3 •
           int t,size;
scanf("%d",&t);
while(t--)
4
 5
 6
 7 ,
                scanf("%d",&size);
for(int i=0;i<size;i++)</pre>
 8
 9
10 1
11
                      for(int j=0;j<size;j++)</pre>
12
                           if((i+j)%2==0)
    printf("W");
else
13
14
15
                                printf("B");
16
17
18
                      printf("\n");
19
20
21
           return 0;
22 }
```

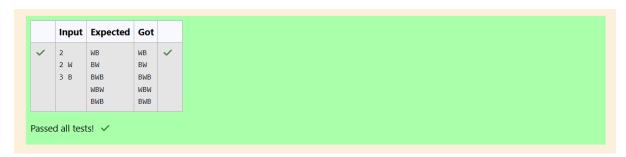


2. Print Our Own Chessboard

Problem statement:

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

```
1 #include<stdio.h>
2 int main()
3 v {
             int t,n;
char ch;
scanf("%d",&t);
while(t--)
 4 5
 6
7
8 v
                    scanf("%d %c",&n,&ch);
for(int i=0;i<n;i++)</pre>
 9
10
11 •
                           for(int j=0;j<n;j++)</pre>
12
13 1
                                 if(ch=='W')
    if((i+j)%2==0)
        printf("W");
    else
14
15
16
17
                                printf("B");
else if((i+j)%2==0)
    printf("B");
else
18
19
20
21
22
                                      printf("W");
23
24
25
                          printf("\n");
26
27
             return 0;
28 }
```



3. pattern painting

Problem statement:

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
```

```
#include<stdio.h>
    #include<string.h>
 3
    int sum(int n)
 4 ₹ {
5
         return n*(n-1)/2;
 6
7 void BSpattern(int N) 8 v {
         int val=0,pthree=0,cnt=0,initial;
char s[100]="**";
9
10
          for(int i=0;i<N;i++)
11
12 v
13
              cnt=0;
14
              if(i>0)
15 1
                  printf("%s",s);
strcat(s,"**");
16
17
18
19
         for(int j=i;j<N;j++)</pre>
20
              if(i>0)
21
22
              {
23
                  cnt++;
24
             printf("%d",++val);
printf("0");
25
26
27
```

```
28
         it (i==0)
29
         {
             int sume=sum(val)*2;
pthree=val+sume+1;
30
31
              initial=pthree;
32
33
34
         initial=initial-cnt;
35
         pthree=initial;
         for(int k=i;k<N;k++)</pre>
36
37
              printf("%d",pthree++);
38
39
              if(k!=N-1)
40
41
                  printf("0");
42
43
         printf("\n");
44
45
46
47
     int main()
48
49
         int N,t;
scanf("%d",&t);
50
51
         for(int i=1;i<=t;i++)
52
53
             scanf("%d",&N);
printf("Case #%d\n",i);
54
55
              BSpattern(N);
56
57
58
         return 0;
59
```



4. Armstrong number

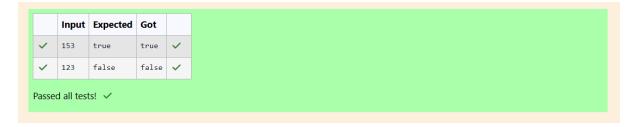
Problem statement:

```
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.
```

Program:

```
int n,org,count=0,sum=0;
scanf("%d",&n);
  6
         org=n;
         while(n>0)
  8
  9
         {
             count++;
n/=10;
 10
 11
 12
         n=org;
while(n>0)
 13
 14
 15
              int t=n%10;
sum+=pow(t,count);
n/=10;
 16
 17
 18
 19
 20
          if(org==sum)
 21 1
 22
              printf("true");
 23
 24
 25
              printf("false");
 26
 27
          return 0;
 28 }
```

Test cases:



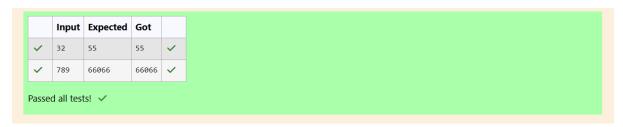
5. Reverse and Add Until Get a Palindrome

Problem statement:

Take a number, reverse it and add it to the original number until the obtained number is a palindrome. Constraints 1<=num<=999999999 Sample Input 1 32 Sample Output 1 55 Sample Input 2 789 Sample Output 2 66066

Program:

```
#include<stdio.h>
    int main()
    long long int n,z,rev,temp1,temp2;
scanf("%lld",&n);
5
        while(1)
6
            temp1=n,rev=0;
            while(n)
10
                rev=rev*10+(n%10);
11
12
               n/=10;
13
14
            z=temp1+rev;
            temp2=z,rev=0;
15
            while(z)
16
17
              rev=rev*10+(z%10);
18
              z=z/10;
19
20
            if(temp2==rev){
21
22
                break;
23
            n=temp2;
24
25
        printf("%lld",temp2);
26
27
        return 0;
28
```



6. Lucky number

Problem statement:

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.	
Here the lucky numbers are 3, 4, 33, 34., and the 3rd lucky number is 33. Sample Input 2:	
Sample Input 2:	
Sample Input 2:	

```
1 #include<stdio.h>
 int islucky(int num){
while(num>0)
 4 v
5
               int digit = num%10;
if(digit!=3&&digit!=4){
    return 0;
}num/=10;
 6 ▼
7
 8
 9
10
          return 1;
11
int findnthlucky(int n)
13 v
14
          int count=0,num=1;
15
          while(1)
16
          {
17
               if(islucky(num))
18 ,
               {
19
                    count++;
20 1
                    if(count==n){
21
                         return num;
22
23
24
               num++;
25
26
     }
int main()
27
28 ₹ {
          int n;
scanf("%d",&n);
printf("%d",findnthlucky(n));
return 0;
29
30
31
32
33 }
```

