WEEK 5

Simple Chessboard

ProblemStatement:

Writeaprogramthatprintsasimplechessboard. Input

format:

Thefirstlinecontainsthenumberofinputs T.

Thelinesafterthatcontainadifferentvalueforsizeofthechessboard Output

format:

Printachessboardofdimensionssize*size.

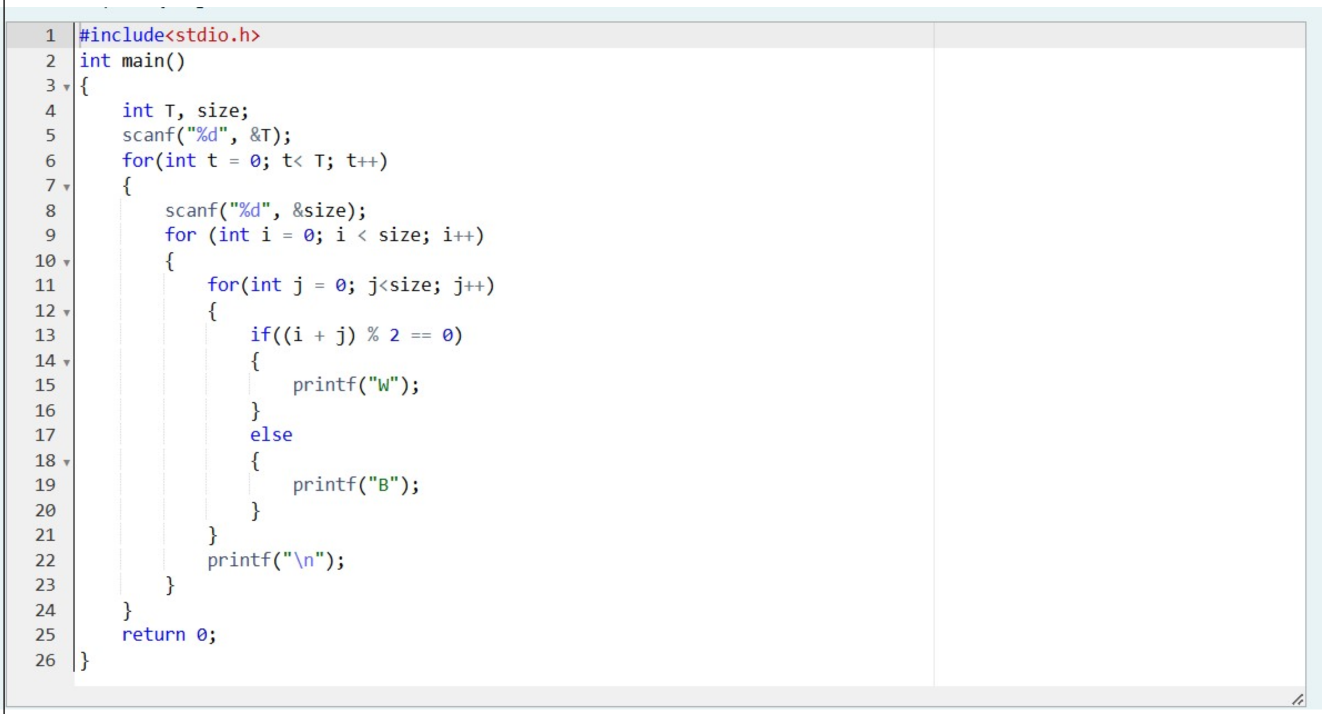
PrintWforwhitespacesandBforblackspaces. Sample

Input:

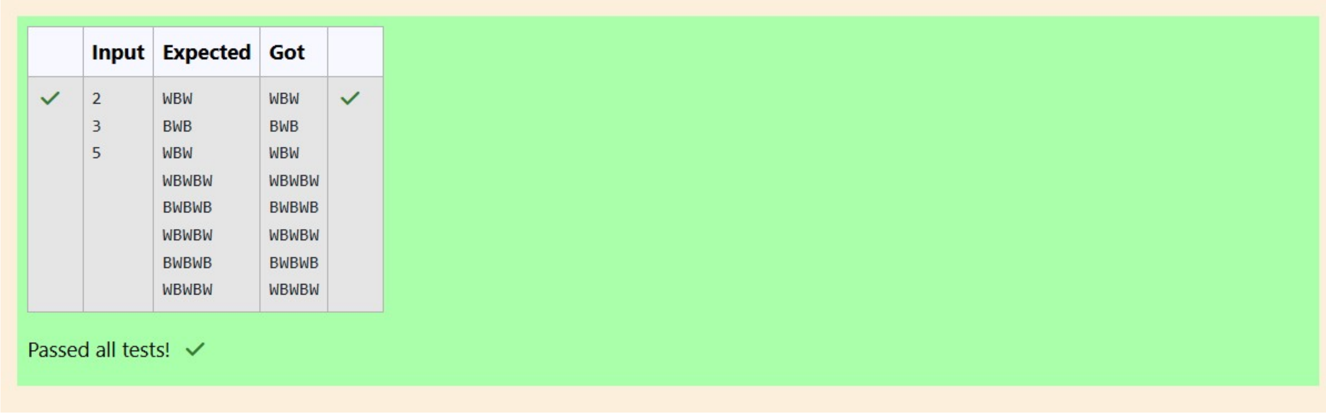
```
2
3
5
```

SampleOutput:

```
WBW
BWB
WBW
WBWBW
BWBWB
WBWBW
BWBWB
WBWBW
```



```c
#include<stdio.h>
int main()
{
    int T, size;
    scanf("%d", &T);
    for(int t = 0; t< T; t++)
    {
        scanf("%d", &size);
        for (int i = 0; i < size; i++)
        {
            for(int j = 0; j<size; j++)
            {
                if((i + j) % 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! ✓

PrintOurOwnChessboard

ProblemStatement:

Let’sprintachessboard!

Writeaprogramthattakesinput:

ThefirstlinecontainsT,thenumberoftestcases

EachtestcasecontainsanintegerNandalsothestartingcharacterofthechessboard Output

Format

Printthechessboardasperthegivenexamples Sample

Input:

2

2W

3B

SampleOutput:

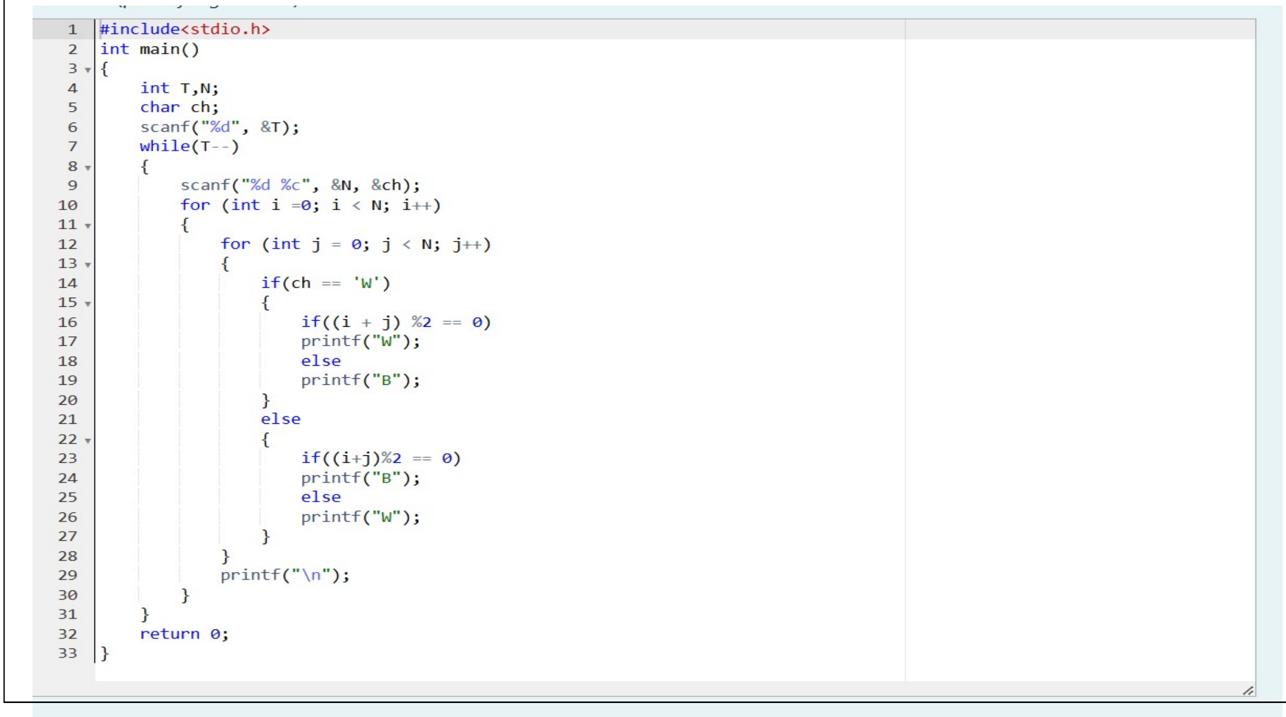
WB

BW

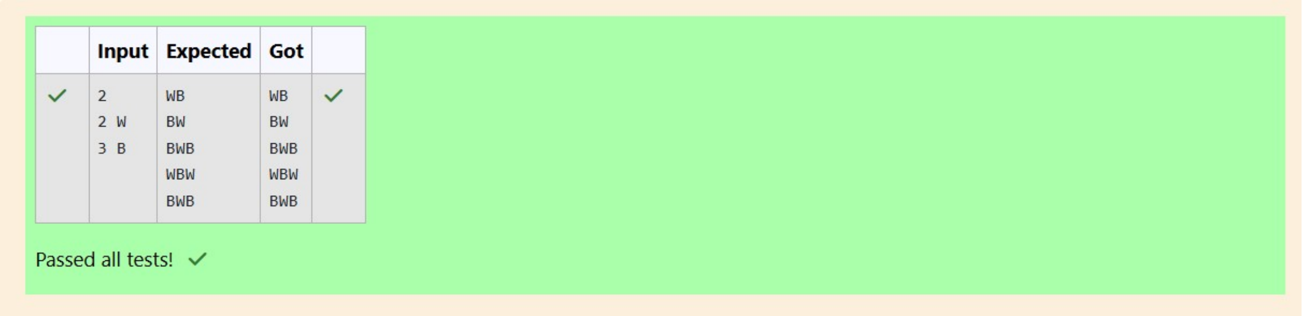
BWB

WBW

BWB



```
#include<stdio.h>
int main()
{
    int T,N;
    char ch;
    scanf("%d", &T);
    while(T--)
    {
        scanf("%d %c", &N, &ch);
        for (int i =0; i < N; i++)
        {
            for (int j = 0; j < N; j++)
            {
                if(ch == 'W')
                {
                    if((i + j) %2 == 0)
                    printf("W");
                    else
                    printf("B");
                }
                else
                {
                    if((i+j)%2 == 0)
                    printf("B");
                    else
                    printf("W");
                }
            }
            printf("\n");
        }
    }
    return 0;
}
```



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

Passed all tests! ✓

PatternPrinting

ProblemStatement:

DecodethelogicandprintthePatternthatcorrespondstogiveninput. If N=

3 then pattern will be:

10203010011012

**4050809

****607

IfN=4,thenpatternwillbe:

1020304017018019020

**50607014015016

****809012013

******10011

Constraints:2<=N<=100 Input

Format

FirstlinecontainsT,thenumberoftestcases,eachtestcasecontainsasingleintegerN Output

Format

FirstlineprintCase#iwhereiisthetestcasenumber,Inthesubsequentline,printthe pattern

SampleInput

3

3

4

5

SampleOutput

Case #1

10203010011012

**4050809

****607

Case #2

1020304017018019020

**50607014015016

****809012013

******10011

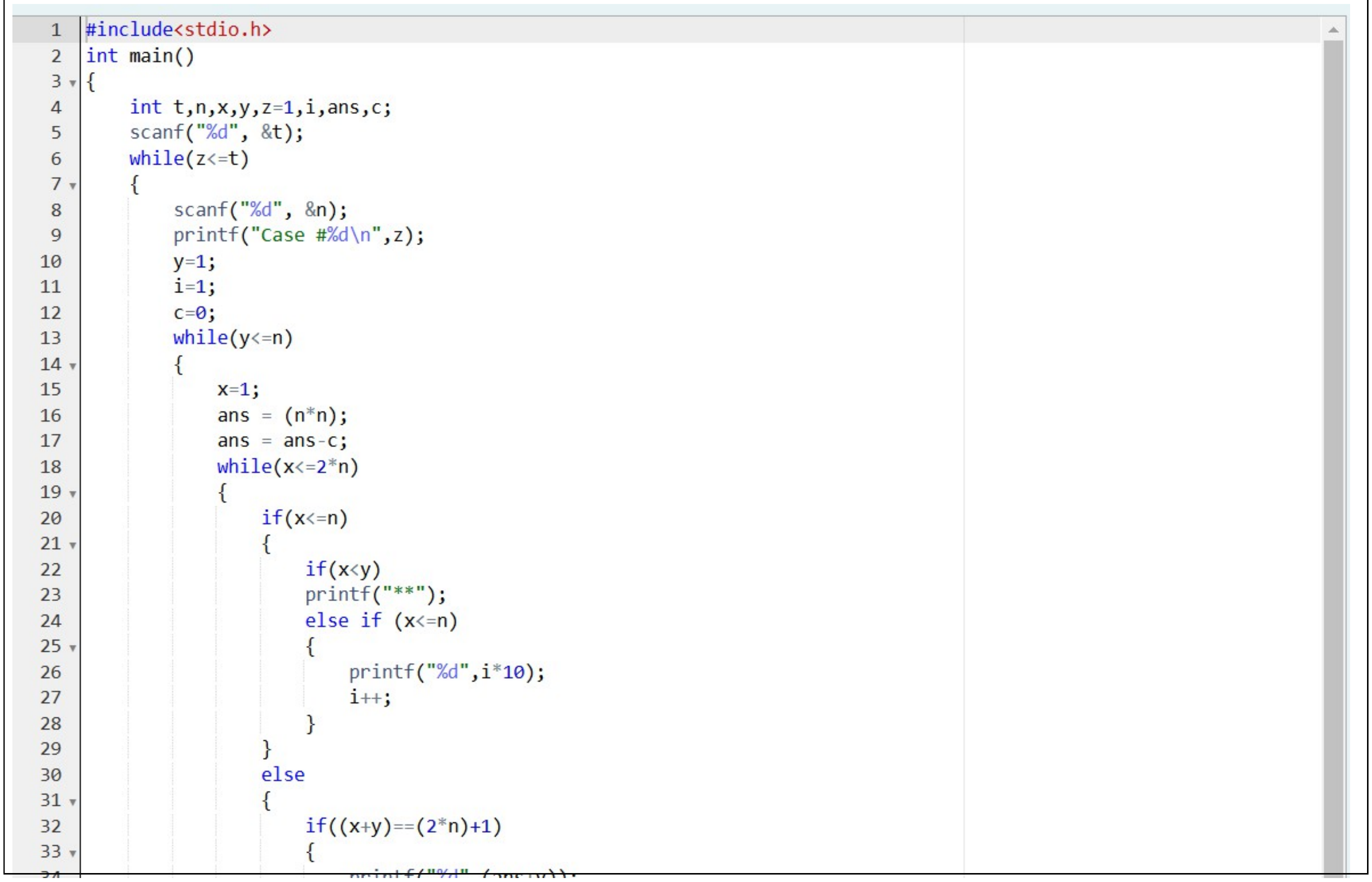
Case #3

102030405026027028029030

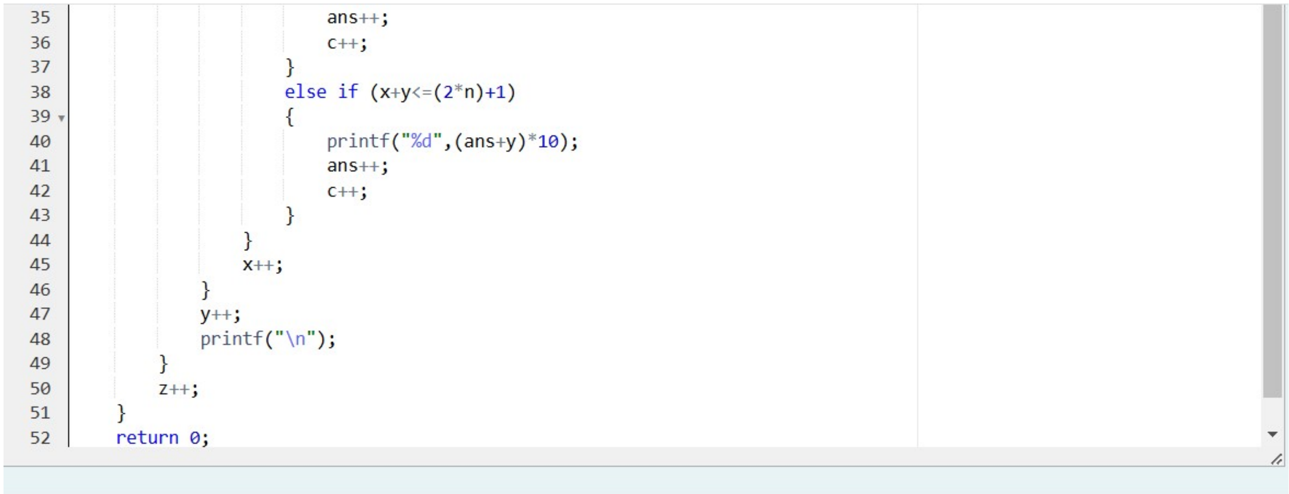
**6070809022023024025

****10011012019020021

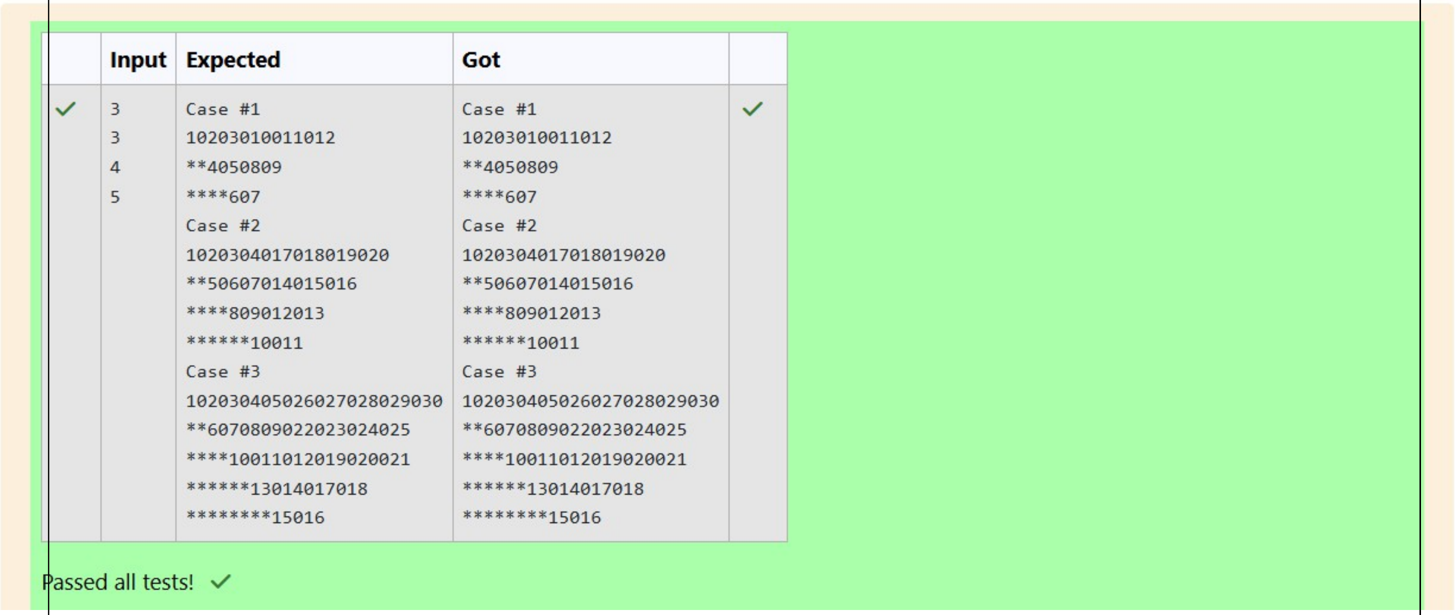
******13014017018



```c
#include<stdio.h>
int main()
{
    int t,n,x,y,z=1,i,ans,c;
    scanf("%d", &t);
    while(z<=t)
    {
        scanf("%d", &n);
        printf("Case #%d\n",z);
        y=1;
        i=1;
        c=0;
        while(y<=n)
        {
            x=1;
            ans = (n*n);
            ans = ans-c;
            while(x<=2*n)
            {
                if(x<=n)
                {
                    if(x<y)
                    printf("**");
                    else if (x<=n)
                    {
                        printf("%d",i*10);
                        i++;
                    }
                }
                else
                {
                    if((x+y)==(2*n)+1)
                    {
                        printf("%d",(ans+y));
```



```
                        ans++;
                        c++;
                    }
                    else if (x+y<=(2*n)+1)
                    {
                        printf("%d",(ans+y)*10);
                        ans++;
                        c++;
                    }
                }
                x++;
            }
            y++;
            printf("\n");
        }
        z++;
    }
    return 0;
```



	Input	Expected	Got	
✓	3 3 4 5	Case #1 10203010011012 **4050809 ****607 Case #2 1020304017018019020 **50607014015016 ****809012013 ******10011 Case #3 102030405026027028029030 **6070809022023024025 ****10011012019020021 ******13014017018 ********15016	Case #1 10203010011012 **4050809 ****607 Case #2 1020304017018019020 **50607014015016 ****809012013 ******10011 Case #3 102030405026027028029030 **6070809022023024025 ****10011012019020021 ******13014017018 ********15016	✓

Passed all tests! ✓

ArmstrongNumber

ProblemStatement:

Thek-digitnumberNisanArmstrongnumberifandonlyifthek-thpowerofeachdigit sums to

N.

GivenapositiveintegerN,returntrueifandonlyifitisanArmstrongnumber. Note: 1 <=

N <= 10^8

Hint:153isa3-digitnumber,and153=1^3+5^3+3^3. Sample

Input:

153

SampleOutput:

true

SampleInput:

123

SampleOutput:

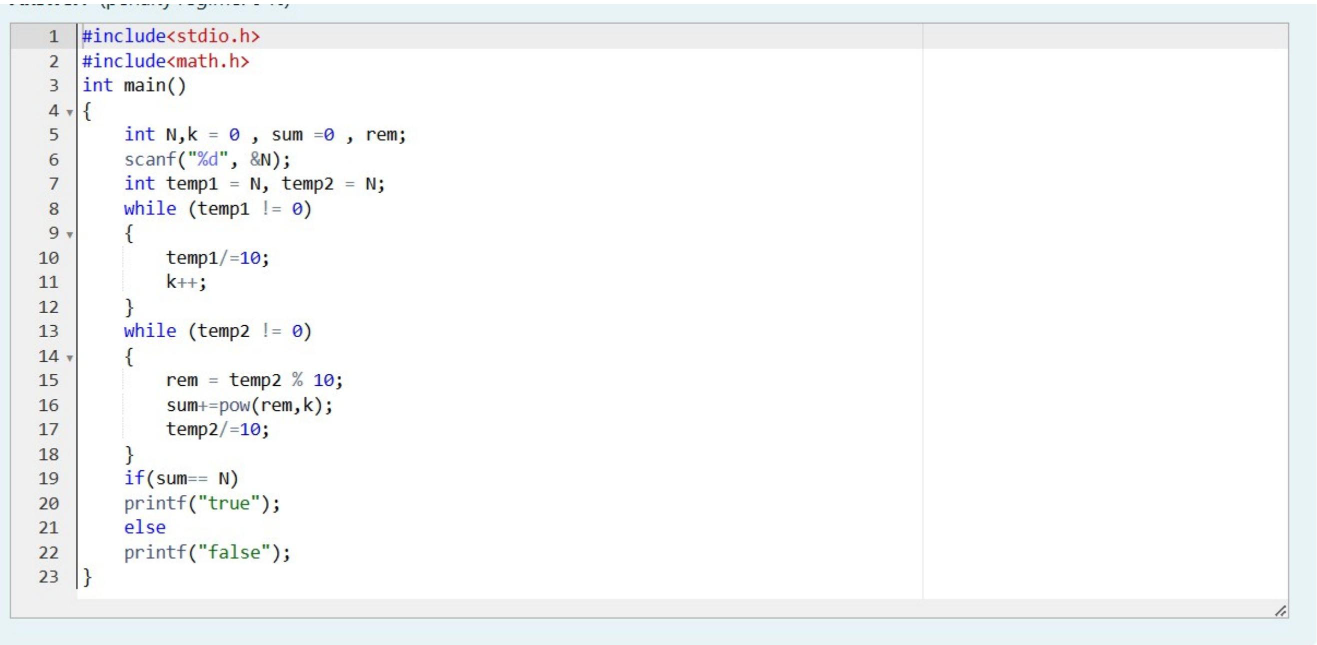
false

SampleInput:

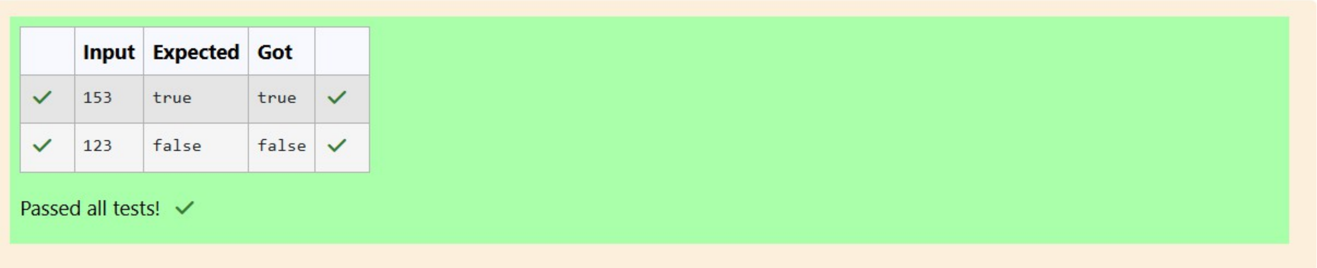
1634

SampleOutput:

true



```c
#include<stdio.h>
#include<math.h>
int main()
{
    int N,k = 0 , sum =0 , rem;
    scanf("%d", &N);
    int temp1 = N, temp2 = N;
    while (temp1 != 0)
    {
        temp1/=10;
        k++;
    }
    while (temp2 != 0)
    {
        rem = temp2 % 10;
        sum+=pow(rem,k);
        temp2/=10;
    }
    if(sum== N)
    printf("true");
    else
    printf("false");
}
```



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

Passed all tests! ✓

ReverseandAddUntilGetaPalindrome

ProblemStatement:

Takeanumber,reverseitandaddittotheoriginalnumberuntiltheobtainednumberis a

palindrome.

Constraints

1<=num<=99999999

SampleInput1

32

SampleOutput1

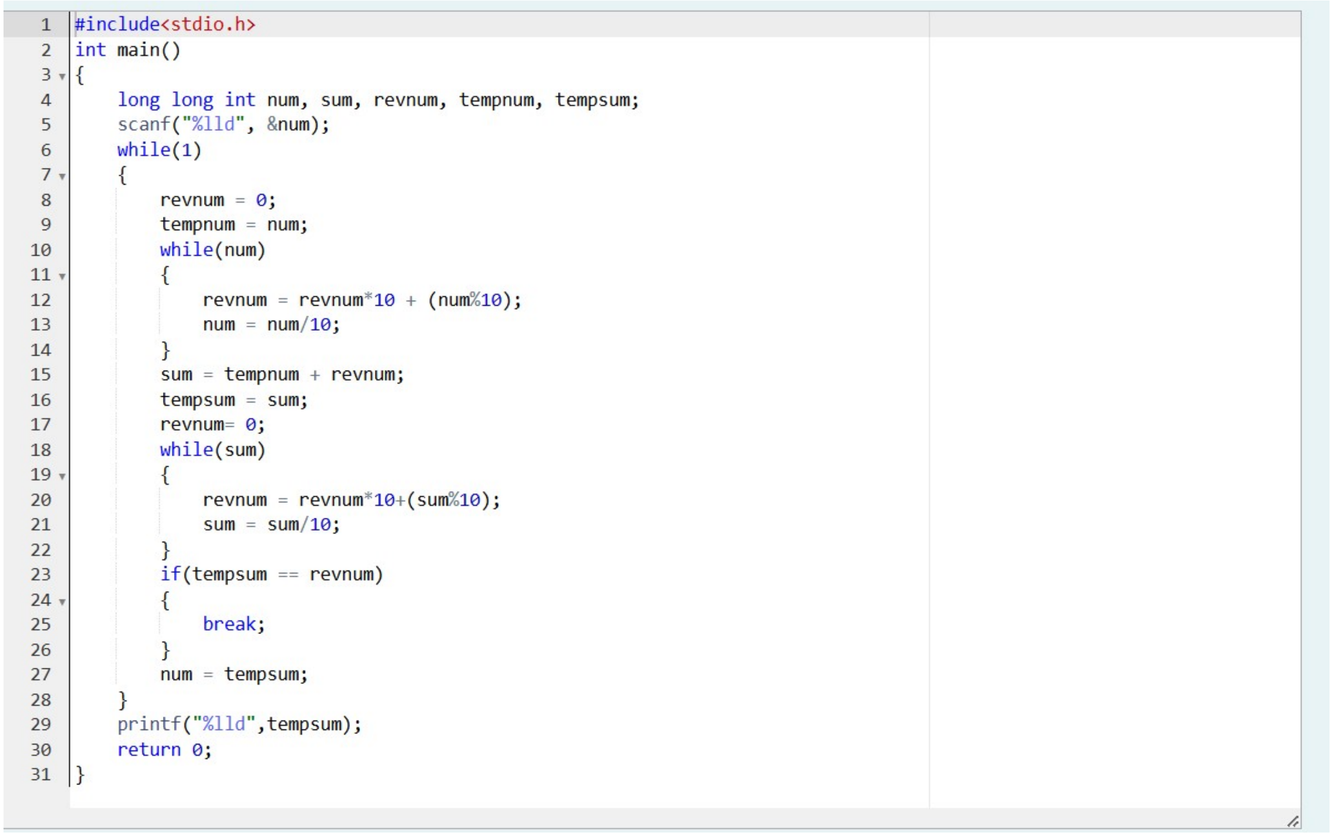
55

SampleInput2

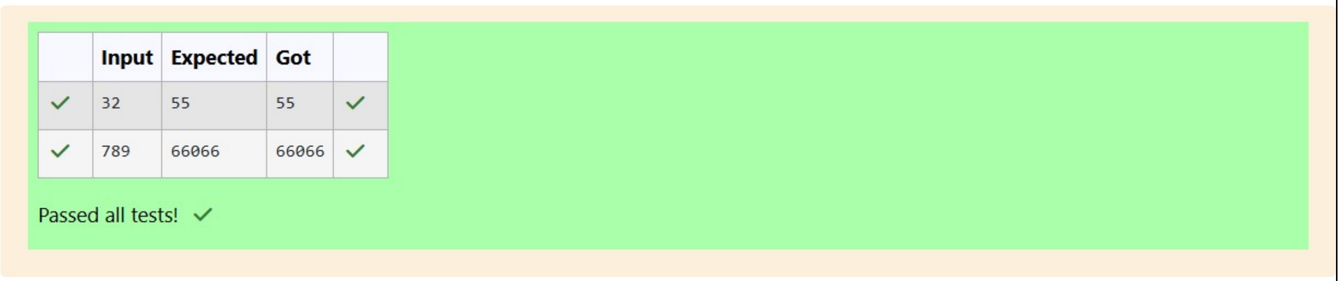
789

SampleOutput2

66066



```c
#include<stdio.h>
int main()
{
    long long int num, sum, revnum, tempnum, tempsum;
    scanf("%lld", &num);
    while(1)
    {
        revnum = 0;
        tempnum = num;
        while(num)
        {
            revnum = revnum*10 + (num%10);
            num = num/10;
        }
        sum = tempnum + revnum;
        tempsum = sum;
        revnum= 0;
        while(sum)
        {
            revnum = revnum*10+(sum%10);
            sum = sum/10;
        }
        if(tempsum == revnum)
        {
            break;
        }
        num = tempsum;
    }
    printf("%lld",tempsum);
    return 0;
}
```



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

Passed all tests! ✓

LuckyNumber

ProblemStatement:

A number is considered lucky if it contains either 3 or 4 or 3 and 4 both in it. Write a programtoprintthenthluckynumber.Example,1stluckynumberis3,and2ndlucky numberis4and3rdluckynumberis33and4thluckynumberis34andsoon.Notethat 13, 40 etc., are not lucky as they have other numbers in it.

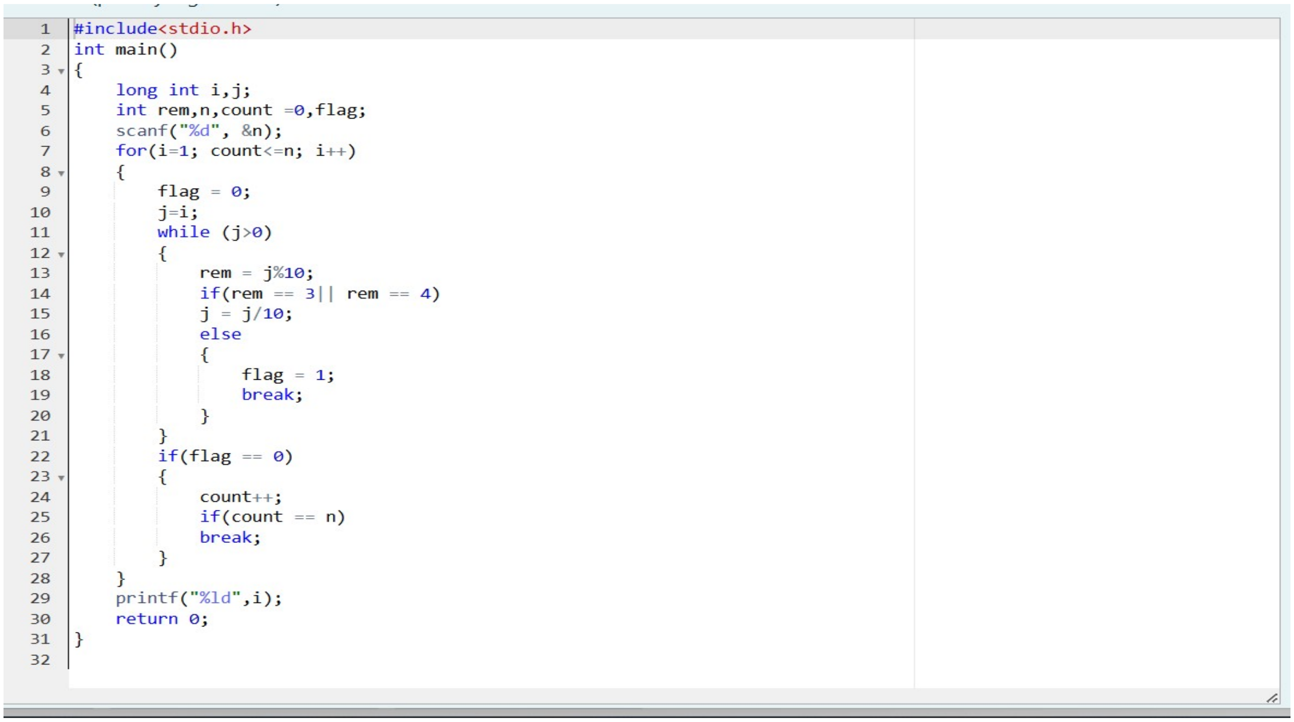
Theprogramshouldacceptanumber'n'asinputanddisplaythenthluckynumberas output.

SampleInput1:

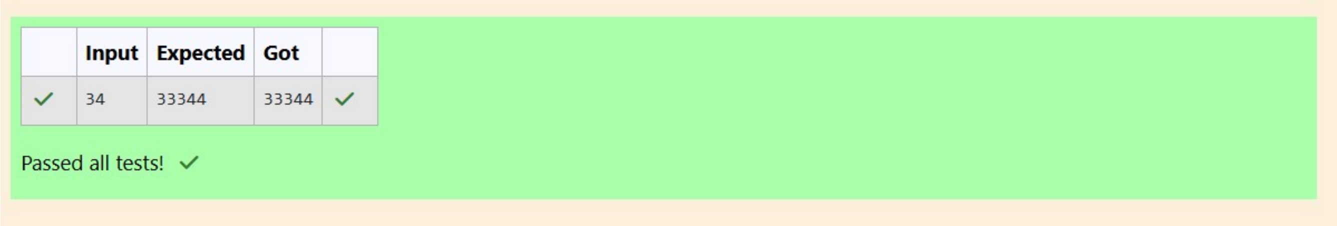
3

SampleOutput1:

33



```
#include<stdio.h>
int main()
{
    long int i,j;
    int rem,n,count =0,flag;
    scanf("%d", &n);
    for(i=1; count<=n; i++)
    {
        flag = 0;
        j=i;
        while (j>0)
        {
            rem = j%10;
            if(rem == 3|| rem == 4)
            j = j/10;
            else
            {
                flag = 1;
                break;
            }
        }
        if(flag == 0)
        {
            count++;
            if(count == n)
            break;
        }
    }
    printf("%ld",i);
    return 0;
}
```



	Input	Expected	Got	
✓	34	33344	33344	✓

Passed all tests! ✓