**Week-05-01**

**1.**

**Question text**

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**Source Code**

#include<stdio.h>

int main()

{

int T%;

scanf("%d", &T);

for (int t=0;t<T;t++)

{

int size;B

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;

}

int size;B

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;

}

**Output**

A screenshot of a computer

Description automatically generated

**2.**

**Question text**

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

**Source Code**

#include<stdio.h>

int main()

{

int T;

scanf("%d", &T);

for (int t=0;t<T;t++) {

int N;

char start;

scanf("%d %c", &N, &start);

char alt= (start = 'W') ? 'B' : 'W';

for (int i=0;i<n;i++)

{

for (int j=0;j<N;j++)

if((i+j) %2 == 0)

{

{

}

printf("%c", start);

}

}

else

{

printf("%c", alt);

}

}

printf("\n");

return 0;

{

for (int j=0;j<N;j++)

{

if((i+j) %2 == 0)

{

printf("%c", start);

}

else

{

printf("%c", alt);

}

}

printf("\n");

}

}

return 0;

}

**Output**

A screenshot of a computer

Description automatically generated

**3.**

**Question text**

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

\*\*\*\*607Case#21020304017018019020

\*\*50607014015016

\*\*\*\*809012013

\*\*\*\*\*\*10011

Case #3

102030405026027028029030

\*\*6070809022023024025

\*\*\*\*10011012019020021

\*\*\*\*\*\*13014017018

\*\*\*\*\*\*\*\*15016

**Source Code**

#include<stdio.h>

int main()

{

int n,v,p3,c, in, i, il, i2, t, ti;

scanf("%d", &t);

for (ti=0;ti<t; ti++)

{

v=0;

scanf("%d", &n);

printf("Case # %d\n",ti+1);

for(i=0;i<n;i++)

{

c=0;

if(i>0)

{

for (i=0;il<i; il++) printf("");

}

for (il=i;il<n;i++)

{

if(i>0) c++;

printf("%d", ++v);

}

if(i==0)

{

p3=v+(v\* (v-1))+1;

in=p3;

}

in-in-c;

p3=in;

for (i2=i;i<n;i++)

{

if(i>0) c++;

printf("%d", ++v);

}

if(i==0)

{

p3=v+(v\* (v-1))+1;

in=p3;

}

in-in-c;

p3=in;

for (i2=i;i<n;i++)

{

printf("%d",p3++);

if(i2!=n-1) printf("0");

}

printf("\n");

}

}

return 0;

}

**Output**

A screenshot of a computer program

Description automatically generated

**Week-05-02**

**1.**

**Question text**

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 != 1^3 + 2^3 + 3^3 = 36

**Example 3:**

**Input:**

1634

**Output:**true

**Note:**

1 <= N <= 10^8**Source Code**

#include<stdio.h>

#include <math.h>

int main()

{

int n,d, nt, AN;

scanf("%d", &n);

nt=n;

for (d=0;nt!=0;d++)

{

nt=nt/10;

}

nt=n;

for(int i=0;i<d; i++)

{

AN=AN+pow(nt %10,d);

nt=nt/10;

}

if(AN==n)

{

printf("true");

}

else

{

printf("false");

return 0;

}

printf("false");

}

return 0;

}

**Output**

A screenshot of a computer

Description automatically generated

**2.**

**Question text**

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**Source Code**

#include<stdio.h>

int main()

{

int num, rev;

int par=0;

scanf("%d", &num);

while(1)

{

par=num;

rev=0;

while(num>0)

{

rev=rev\*10+num%10;

num/=10;

}

num=par+rev;

int temp=num, 1=0;

while(temp>0)

{

int digit-temp%10;

1-1\*10+digit;

temp/=10;

}

if (num==1)

{

break;

}

}

printf("%d", num);

return 0;

num/=10;

}

num=par+rev;

int temp=num, 1=0;

while(temp>0)

{

int digit-temp%10;

1=1\*10+digit;

temp/=10;

}

if (num==1)

{

break;

}

}

printf("%d", num);

return 0;

}

**Output**

A screenshot of a test

Description automatically generated

**3.**

**Question text**

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**Source Code**

#include<stdio.h>

int is lucky (int num)

{

while (num>0)

{

int num1;

num1 = num%10;

if (num1!=3&&num1!=4)

return 0;

num/=10;

}

return 1;

}

int main()

{

int n,c=0, num = 1;

scanf("%d", &n);

while(c<n)

if(islucky (num))

{

{

c++;

if(c==n)

{

printf("%d", num);

break;

}

}

num++;

{

int n,c=0, num = 1;

scanf("%d", &n);

while(c<n)

{

if(islucky (num))

{

c++;

if(c==n)

{

printf("%d", num);

break;

}

}

num++;

}

return 0;

}

**Output**

A screenshot of a computer

Description automatically generated