

# Week 05

Question 1:

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.

Program:

Attempt 1

<b>Status</b>	Finished
<b>Started</b>	Monday, 23 December 2024, 5:33 PM
<b>Completed</b>	Monday, 9 December 2024, 1:32 AM
<b>Duration</b>	14 days 16 hours

wer: (penalty regime: 0 %)

```

1 #include<stdio.h>
2 int main(){
3     int t;
4     scanf("%d",&t);
5     for(int k=0;k<t;k++){
6         int n;
7         char p;
8         scanf("%d",&n);
9         for(int row = 0; row < n;row++){
10             p=(row%2==0)? 'W':'B';
11             for(int col=0;col<n;col++){
12                 printf("%c",p);
13                 p=(p=='W')?'B':'W';
14             }
15             printf("\n");
16         }
17     }
18     return 0;
19 }
```

Output:

	Input	Expected	Got	
✓	2	WBW	WBW	✓
	3	BWB	BWB	
	5	WBW	WBW	
		WBWBW	WBWBW	
		BWBWB	BWBWB	
		WBWBW	WBWBW	
		BWBWB	BWBWB	
		WBWBW	WBWBW	

Passed all tests! ✓

Question 2:

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 1:

2

2 W

3 B

Sample Output 1:

WB

BW

BWB

WBW

BWB

Program:

Attempt 1

<b>Status</b>	Finished
<b>Started</b>	Monday, 23 December 2024, 5:33 PM
<b>Completed</b>	Monday, 9 December 2024, 1:32 AM
<b>Duration</b>	14 days 16 hours

answer: (penalty regime: 0 %)

```

1  #include<stdio.h>
2  int main(){
3      int t;
4      scanf("%d",&t);
5      for(int x=1;x<=t;x++){
6          int n;
7          char c,p;
8          scanf("%d %c",&n,&c);
9          p=c;
10         for(int row =0;row<n;row++){
11             if(row%2==0){
12                 p=c;
13             }
14             else{
15                 p=(c=='W')?'B':'W';
16             }
17             for(int col=1;col<=n;col++){
18                 printf("%c",p);
19                 p = (p == 'W')?'B':'W';
20             }
21             printf("\n");
22         }
23     }
24     return 0;
25 }
```

99  
Output:

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

Passed all tests! ✓

Question 3:

Decode the logic and print the Pattern that corresponds to given input.

If N= 3 then pattern will be :

10203010011012

\*\*4050809

\*\*\*\*60

If N= 4, then pattern will be:

1020304017018019020

\*\*50607014015016

\*\*\*\*809012013

\*\*\*\*\*10011

Input Format:

First line contains T, the number of test cases

Each test case contains a single integer N

Attempt 1

<b>Status</b>	Finished
<b>Started</b>	Monday, 23 December 2024, 5:33 PM
<b>Completed</b>	Monday, 9 December 2024, 1:32 AM
<b>Duration</b>	14 days 16 hours

Program

```
1 #include<stdio.h>
2 int main(){
3     int t;
4     scanf("%d",&t);
5     for(int x=1;x<=t;x++){
6         printf("Case #%d\n",x);
7         int n;
8         scanf("%d",&n);
9         int f=1,b=n*(n+1);
10        for(int i=0;i<n;i++){
11            for(int k=0;k<2*i;k++){
12                printf("*");
13            }
14            printf("%d",f);
15            f++;
16            for(int j=2;j<=n-i;j++){
17                printf("0%d",f);
18                f++;
19            }
20            for(int l=b-(n-i)+1;l<=b;l++){
21                printf("0%d",l);
22            }
23            b-=n-i;
24            printf("\n");
25        }
26    }
27    return 0;
28 }
29 }
```

Output:

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

Passed all tests! ✓

#### Question 4:

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.

Sample Input 1:

153

Sample Output 1:

true

Program:

Attempt 1

**Status** Finished

**Started** Monday, 23 December 2024, 5:33 PM

**Completed** Monday, 9 December 2024, 1:32 AM

**Duration** 14 days 16 hours

Answer: (penalty regime: 0 %)

```

1  #include<stdio.h>
2  #include<math.h>
3  int main(){
4      int n,k=0,sum=0;
5      scanf("%d",&n);
6      int temp=n;
7      while(temp>0){
8          k++;
9          temp/=10;
10     }
11     temp=n;
12     while(temp>0){
13         sum+=pow(temp%10,k);
14         temp/=10;
15     }
16     if(sum==n){
17         printf("true");
18     }
19     else{
20         printf("false");
21     }
22     return 0;
23 }
```

Output:

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

Passed all tests! ✓



Question 5:

Take a number, reverse it and add it to the original number until the obtained number is a palindrome.

Sample Input 1 :

32

Sample Output 1 :

55

Attempt 1

<b>Status</b>	Finished
<b>Started</b>	Monday, 23 December 2024, 5:33 PM
<b>Completed</b>	Monday, 9 December 2024, 1:32 AM
<b>Duration</b>	14 days 16 hours

Program

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int reverse(int num);
3 int main(){
4     int n;
5     scanf("%d",&n);
6     while(n!=reverse(n)){
7         n=(n+reverse(n));
8     }
9     printf("%d",n);
10 }
11 int reverse(int num){
12     int rev_num=0;
13     while(num>0){
14         rev_num=(rev_num*10+num%10);
15         num/=10;
16     }
17     return rev_num;
18 return 0;
19 }
```

Output:

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

Your code failed one or more hidden tests.

Your code must pass all tests to earn any marks. Try again.

Question 6:

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

Program:

Attempt 1	
<b>Status</b>	Finished
<b>Started</b>	Monday, 23 December 2024, 5:33 PM
<b>Completed</b>	Monday, 9 December 2024, 1:32 AM
<b>Duration</b>	14 days 16 hours

```
1 #include<stdio.h>
2 #include<math.h>
3 int main(){
4     int n,k=1,num=0;
5     scanf("%d",&n);
6     while(n>0){
7         if(n-pow(2,k)>=0){
8             n=n-pow(2,k);
9             k++;
10        }
11        else{
12            break;
13        }
14    }
15    for(int i=0;i<k;i++){
16        int t=0,p=3;
17        for(int j=1;j<=n;j++){
18            if(t==pow(2,i)){
19                t=1;
20                p=(p==3)? 4:3;
21            }
22            else{
23                t++;
24            }
25        }
26        num= p*pow(10,i)+num;
27    }
28    printf("%d",num);
29    return 0;
}
```

Output:

	Input	Expected	Got	
✓	34	33344	33344	✓

Passed all tests! ✓