WEEK 5[SESSION 1]

Question **1**

Correct

Marked out of 3.00

Flag question

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.

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int t;
    scanf("%d",&t);
    while(t-->0){
        int a;
        scanf("%d",&a);
        for(int i=1;i<=a;i++){
            for(int j=1;j<=a;j++){
                if(((i%2==0)&&(j%2==0))||((i%2==1)&&(j%2==1))){
                    printf("%c",'W');
                }else{
                    printf("%c",'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! ✓

Question **2**

Correct

Marked out of 5.00

Flag question

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

```c
#include<stdio.h>
int main(){
    int t;
    scanf("%d",&t);
    while(t-->0){
        int a;
        char b;
        scanf("%d %c",&a,&b);
        for(int i=1;i<=a;i++){
            if(b=='W'){
                for(int j=1;j<=a;j++){
                    if((i%2==0&&j%2==0)||(i%2==1&&j%2==1)){
                        printf("%c",'W');
                    }else{
                    printf("%c",'B');
                }

                }
                printf("\n");
                }else{
                for(int j=1;j<=a;j++){
                    if((i%2==0&&j%2==0)||(i%2==1&&j%2==1)){
                        printf("%c",'B');
                    }else{
                        printf("%c",'W');
                    }

                }
                printf("\n");
            }
        }

    }

```

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

Passed all tests! ✓

Question **3**

Correct

Marked out of 7.00

Flag question

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

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int t;
    scanf("%d",&t);
    for(int x=1;x<=t;x++){
        printf("Case #%d\n",x);
        int n;
        scanf("%d",&n);
        int f=1;
        int b=n*(n+1);
        for(int i=0;i<n;i++){
            for(int k=0;k<2*i;k++){
                printf("*");
            }
            printf("%d",f);
            f++;
            for(int j=2;j<=n-i;j++){
                printf("0%d",f);
                f++;
            }
            for(int l=b-(n-i)+1;l<=b;l++){
                printf("0%d",l);
    }
    b-=n-i;
    printf("\n");
}
}
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! ✓

WEEK 5[SESSION 2]

Question **1**

Correct

Marked out of 3.00

Flag question

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.

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
#include<math.h>
int main(){
    int a,count=0,sum=0;
    scanf("%d",&a);
    int n=a;
    while(n!=0){
        count++;
        n/=10;
    }
    int c=a;
    while(c>0){
        int m=c%10;
        sum+=pow(m,count);
        c/=10;
    }
    if(sum==a){
        printf("true");
    }else{
        printf("false");
    }
    return 0;
}
```

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

Passed all tests! ✓

Question **2**

Correct

Marked out of 5.00

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 %)

```c
#include<stdio.h>
int main(){
    int a,sum=0,rev,i=0;
    scanf("%d",&a);
    do{
        sum=a;
        rev=0;
    while(a>0){
        int m=a%10;
        rev=(rev*10)+m;
        a/=10;
    }
    a=sum+rev;
    i++;
    }while(rev!=sum||i==1);
    printf("%d",rev);
    return 0;
}
```

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

Question **3**

Correct

Marked out of 7.00

Flag question

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.

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main(){
    int n,valid=0,i=0,a=1,temp;
    scanf("%d",&n);
    while(i<n){
        temp=a;
        while(temp>0){
            valid=0;
            if(temp%10!=3&&temp%10!=4){
                valid=1;
                break;
            }
            temp/=10;
        }
        if(valid==0){
            i++;
        }
        a++;
    }
    printf("%d",--a);
    return 0;
}
```

	Input	Expected	Got	
✓	34	33344	33344	✓