

Week-05-Nested Loops - while and for, Jumps in Loops

Week-05-01-Practice Session-Coding

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Tuesday, 3 December 2024, 2:25 PM
Duration	20 days 3 hours

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 different values for size of the chessboard

Output format:

Print a chessboard of dimensions size * size. Print a W for white spaces and B for black spaces.

Input:

2
3
5

Output:

WBW
BWB
WBW
WBWBW
BWBWB
WBWBW
BWBWB
WBWBW

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main(){
3     int T,d,i=0,a,b,o;
4     char c;
5     scanf("%d",&T);
6     while(i<T){
7         scanf("%d",&d);
8         a=0;
9         while(a<d){
10            {
11                o=1;
12                b=0;
13                if(a%2==0)
14                {
15                    o=0;
16                }while(b<d)
17                {c='B';
18                if(b%2==o)
19                {
20                    c='W';
21                }
```

```

22     printf("%c",c);
23     b++;
24     } a+=1;
25     printf("\n");
26   }
27   i=i+1;
28 }
29 }
```

	Input	Expected	Got	
✓	2 3 5	WBW BwB WBW WBWB BwBWB WBWBW WBWBW BwBWB WBWBW	WBW BwB WBW WBWB BwBWB WBWBW 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

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
```

Answer: (penalty regime: 0 %)

```

1 #include<stdio.h>
2 int main(){
3     int T,d,i,a,b,o,z;
```

```

4     char c,s;
5     scanf("%d",&T);
6     for(i=0;i<T;i++)
7     {
8         scanf("%d %c",&d,&s);
9         for(a=0;a<d;a++)
10        {
11            z=(s=='W')? 0:1;
12            o=(a%2==z)? 0:1;
13            for(b=0;b<d;b++)
14            {
15                c=(b%2==o)? 'W': 'B';
16                printf("%c",c);
17            }
18            printf("\n");
19        }
20    }
21    return 0;
22 }
```

	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
```

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
```

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main(){
3     int n,v,p3,c,in,i,i1,i2,t,ti;
4     scanf("%d",&t);
5     for(ti=0;ti<t;ti++){
6         v=0;
7         scanf("%d",&n);
8         printf("Case #%d\n",ti+1);
9         for(i=0;i<n;i++){
10             c=0;
11             if(i>0){
12                 for(i1=0;i1<i;i1++) printf("**");
13             }
14             for(i1=i;i1<n;i1++){
15                 if(i>0) c++;
16                 printf("%d0",++v);
17             }
18             if(i==0){
19                 p3=v*(v*(v-1))+1;
```

```

20     in=p3;
21 }
22
23 p3=in;
24 for(i2=i;i2<n;i2++){
25     printf("%d",p3++);
26     if(i2!=n-1) printf("0");
27 }printf("\n");
28
29 }
30
31 }
32
33

```

	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	✓

Week-05-02-Practice Session-Coding

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Friday, 6 December 2024, 9:18 AM
Duration	17 days 8 hours

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.

Example 1:

Input:

153

Output:

true

Explanation:

153 is a 3-digit number, and $1^3 + 5^3 + 3^3 = 153$.

Example 2:

Input:

123

Output:

false

Explanation:

123 is a 3-digit number, and $1^3 + 2^3 + 3^3 = 36$.

Example 3:

Input:

1634

Output:

true

Note:

$1 \leq N \leq 10^8$

Answer: (penalty regime: 0 %)

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

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

Question 2

Correct

Marked out of
5.00

[Flag question](#)

Take a number, reverse it and add it to the original number until the obtained number is a palindrome. Constraints $1 \leq \text{num} \leq 99999999$ Sample Input 1 32 Sample Output 1 55 Sample Input 2 789 Sample Output 2 66066

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 v int main(){
3     int r,n,nt=0,i=0;
4     scanf("%d",&n);
5 v     do{ nt=n,r=0;
6         while(n!=0){
7             r=r*10+n%10;
8             n=n/10;
9         }n=nt+r;
10        i++;
11    }
12    while(r!=nt || i==1);
13    printf("%d",r);
14    return 0;
15 }
```

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

Passed all tests! ✓

Question ✓
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.

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:

Sample Output 2:

33344

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main(){
3     int n=1,i=0,nt,co=0,e;
4     scanf("%d",&e);
5     while(i<e)
6     {
7         nt=n;
8         while(nt!=0)
9         { co=0;
10            if(nt%10!=3 && nt%10!=4)
11            {
12                co=1;
13                break;
14            }nt=nt/10;
15        }
16        if(co==0)
17        {
18            i++;
19        } n++;
20    }
```

```
-- 21 | printf("%d",--n);
22 }
```

	Input	Expected	Got	
✓	34	33344	33344	✓

Passed all tests! ✓