RITTVIK S 2024-CSE 2116240701616

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

Week-05-01-Practice Session-Coding

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.

Source code

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;
}
```

Output

	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! ✓

Result

The above program is executed successfully and provides the above output.

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

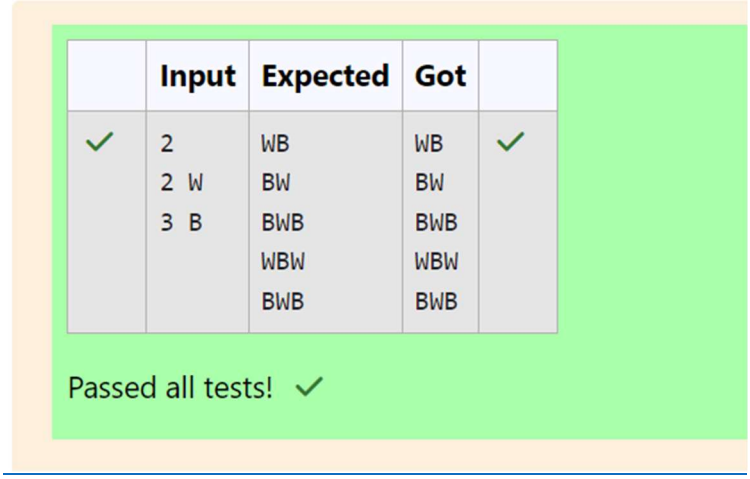
Each test case contains an integer N and also the starting character of the chessboard

Source code

Answer: (penalty regime: 0 %)

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

Output



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

Passed all tests! ✓

Result

The above program is executed successfully and provides the above output.

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

Source code

```c
#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(il=0;il<i;il++) printf("**");
            }
        for(il=i;il<n;il++){
            if(i>0)  c++;
            printf("%d0",++v);
        }
        if(i==0){
            p3=v+(v*(v-1))+1;
            in=p3;
        }
        in=in-c;
        p3=in;
        for(i2=i;i2<n;i2++){
            printf("%d",p3++);
            if(i2!=n-1) printf("0");
        }printf("\n");
        }
        
        
    }
}
```

Output

	Input	Expected	Got	
✓	3 3 4 5	Case #1 10203010011012 **4050809 ****607 Case #2 1020304017018019020 **50607014015016 ****809012013 ******10011 Case #3 10203040502602702802903 0 **6070809022023024025 ****10011012019020021 ******13014017018 ********15016	Case #1 10203010011012 **4050809 ****607 Case #2 1020304017018019020 **50607014015016 ****809012013 ******10011 Case #3 10203040502602702802903 0 **6070809022023024025 ****10011012019020021 ******13014017018 ********15016	✓

Passed all tests! ✓

Result

The above program is executed successfully and provides the above output.

Week-05-02-Practice Session-Coding

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

Source code

```c
#include <stdio.h>
#include <math.h>

int main()
{
    int n, org, count=0, sum =0 ;
    scanf("%d",&n);
    org = n;
    while(n >0)
    {
        count ++;
        n/=10;
    }
    n = org;
    while (n>0)
    {
     int t = n%10;
     sum+= pow(t,count);
     n/=10;
    }
    if(org==sum)
    {
        printf("true");
    }
   else{
       printf("false");
   }
   return 0;
}
```

Output

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

Passed all tests! ✓

Result

The above program is executed successfully and provides the above output.

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

Source code

```c
#include <stdio.h>

int main()
{
    long long int n,z,rev,temp1,temp2;
    scanf("%lld", &n);
    while(1)
    {
        temp1=n, rev=0;
        while(n)
        {
            rev = rev*10 +(n%10);
            n/=10;
        }
        z = temp1 + rev;
        temp2 = z, rev =0;
        while(z)
        {
            rev = rev*10 +(z%10);
            z = z /10;
        }
        if (temp2 == rev) {
            break;
        }
        n = temp2;
    }
    printf("%lld", temp2);
    return 0;
}
```

Output

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

Passed all tests! ✓

Result

The above program is executed successfully and provides the above output.

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.

Source code

```c
#include <stdio.h>

int islucky(int num) {
    while (num>0)
    {
        int digit = num%10;
        if(digit!=3&&digit!=4)
        {
            return 0;
        }
        num /=10;
    }
    return 1;
    }

    int findnthlucky(int n)
    {
        int count =0, num=1;
        while(1)
        {
            if (islucky(num))
            {
                count ++;
                if (count ==n){
                    return num;
                }
            }
            num++;
        }
    }
```

Output

	Input	Expected	Got	
✓	34	33344	33344	✓

Passed all tests! ✓

Result

The above program is executed successfully and provides the above output.