

REC-CIS

GE23131-Programming Using C-2024

Quiz navigation



Show one page at a time

Finish review

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Thursday, 28 November 2024, 5:17 PM
Duration	25 days

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:

REC-CIS

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

REC-CIS

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int T,d,i=0,i1,i2,o;
5     char c;
6     scanf("%d",&T);
7     while(i<T)
8     {
9         scanf("%d",&d);
10        i1=0;
11        while(i1<d)
12        {
13            o=1;
14            i2=0;
15            if(i1%2==0)
16            {
17                o=0;
18            }
19            while(i2<d)
20            {
21                c='B';
22                if(i2%2==o)
23                {
24                    c='W';
25                }
26                printf("%c",c);
27                i2++;
28            }
29            i1+=1;
```


4



Review

Review

REC-CIS

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

Sample Input / Output

Input:

2

2 W

3 B

REC-CIS

Output:

WB

BW

BWB

WBW

BWB

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int T,d,i,i1,i2,o,z;
5     char c,s;
6     scanf("%d",&T);
7     for(i=0;i<T;i++)
8     {
9         scanf("%d %c",&d,&s);
10        for(i1=0;i1<d;i1++)
11        {
12            z=(s=='W')?0:1;
13            o=(i1%2==z)?0:1;
14            for(i2=0;i2<d;i2++)
15            {
```

REC-CIS

```

16         c=(i2%2==0)?'W' : 'B';
17         printf("%c",c);
18     }
19     printf("\n");
20 }
21 }
22 return 0;
23 }
    
```

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

Passed all tests! ✓

Question **3**

Correct

Marked out of
Review

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

REC-CIS

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

REC-CIS

 $2 \leq N \leq 100$

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

REC-CIS

Output

Case #1

10203010011012

**4050809

****607

Case #2

1020304017018019020

**50607014015016

****809012013

*****10011

Case #3

102030405026027028029030

**6070809022023024025

****10011012019020021

*****13014017018

*****15016

REC-CIS

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++){
13                     printf("**");
14                 }
15                 for(i1=i;i1<n;i1++){
16                     if(i>0)c++;
17                     printf("%d0",++v);
18                 }
19                 if(i==0){
20                     p3=v+(v*(v-1))+1;
21                     in=p3;
22                 }
23                 in=in-c;
24                 p3=in;
25                 for(i2=i;i2<n;i2++){
26                     printf("%d",p3++);
27                     if(i2!=n-1)
28                         printf("0");
29                 }printf("\n");

```


4



REC-CIS

GE23131-Programming Using C-2024

Quiz navigation

[Show one page at a time](#)[Finish review](#)

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Thursday, 5 December 2024, 11:02 AM
Duration	18 days 6 hours

Question **1**

Not answered

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

REC-CIS

Output:

true

Explanation:

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

Example 2:

Input:

123

Output:

false

Explanation:

REC-CIS

Explanation:

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

Example 3:

Input:

1634

Output:

true

Note:

$1 \leq N \leq 10^8$

REC-CIS

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 #include<math.h>
3 int main()
4 {
5     int n;
6     scanf("%d",&n);
7     int x=0,n2=n;
8     while (n2!=0)
9     {
10         x++;
11         n2=n2/10;
12     }
13     int sum=0;
14     int n3=n,n4;
15     while (n3!=0)
16     {
17         n4=n3%10;
18         sum=sum+pow(n4,x);
19         n3=n3/10;
20     }
21     if(n==sum)
22     {
23         printf("true");
24     }
25     else
26     {
27         printf("false");
28     }
29     return 0;
```


REC-CIS

30 | }

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

Passed all tests! ✓

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 int main()
3 {
4     int rn,n,nt=0,i=0;
5     scanf("%d",&n);
6     do{
```

⚠ Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=108257&cmid=123

REC-CIS

```
7      nt=n;rn=0;
8      while(n!=0)
9      {
10         rn=rn*10+n%10;
11         n=n/10;
12     }
13     n=nt+rn;
14     i++;
15 }
16 while(rn!=nt || i==1);
17 printf("%d",rn);
18 return 0;
19 }
```

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

Passed all tests! ✓

REC-CIS

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.

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.

⚠ Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=108257&cmid=123

REC-CIS

Sample Input 2:

34

Sample Output 2:

33344

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int n=1,i=0,nt,co=0,e;
5     scanf("%d",&e);
6     while(i<e)
7     {
8         nt=n;
```

REC-CIS

```
9      while(nt!=0)
10     {
11         co=0;
12         if(nt%10!=3&&nt%10!=4)
13         {
14             co=1;
15             break;
16         }
17         nt=nt/10;
18     }
19     if(co==0)
20     {
21         i++;
22     }
23     n++;
24 }
25 printf("%d",--n);
26 return 0;
27 }
```

	Input	Expected	Got	
✓	34	33344	33344	✓

Passed all tests! ✓