

REC-CIS

Sample Input / Output

Input:

2

2 W

38

Output:

WB

BW

BWB

WBW

BWB

Answer: (penalty regime: 0 %)

```

1 #include<stdio.h>
2 int main()
3 {
4     int T,t,l,k;
5     char c;
6     scanf("%d",&t);
7     for(i=0;i<t;i++)
8     {
9         scanf("%d %c",&t,&c);
10        for(int j=0;j<t;j++)
11        {
12            int s=c%2;
13            if(s==0)
14            {
15                for(k=0;k<t;k++)
16                {
17                    if((j+k)%2==0)
18                        printf("B");
19                }
20            }
21        }
22    }
23 }

```

47

	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
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

```
9 scanf("%d %c",&t,&c);
10 for(int j=0;j<t;j++)
11 {
12     int s=c%2;
13     if(s==0)
14     {
15         for(k=0;k<t;k++)
16         {
17             if((j+k)%2==0)
18             {
19                 printf("B");
20             }
21             else
22             {
23                 printf("W");
24             }
25         }
26     }
27     else
28     {
29         for(k=0;k<t;k++)
30         {
31             if((j+k)%2==0)
32             {
33                 printf("W");
34             }
35             else
36             {
37                 printf("B");
38             }
39         }
40         printf("\n");
41     }
42 }
43 return 0;
44 }
45
46
47
```

Input	Expected	Got	
2	WB	WB	
2 W	BW	BW	

REC-CIS

```
**50607014015016
```

```
****809012013
```

```
*****10011
```

Constraints

$$2 \leq N \leq 100$$

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
```

Output

Case #1

```
10203010011012
```

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 {
4     int n,v,p3,c,in,i,i1,i2,t,ti;
5     scanf("%d",&t);
6     for(ti=0;ti<t;ti++){
7         v=0;
8         scanf("%d",&n);
9         printf("Case #\n",ti+1);
10        for(i=0;i<n;i++){
11            c=0;
12            if(i>0){
13                for(i1=0;i1<i;i1++)printf(" ");
14            }
15        }
16    }
```

REC-CIS

GE23131-Programming Using C-2024

Quiz navigation

1

2

3

Show one page at a time

Finish review

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Tuesday, 10 December 2024, 4:12 PM
Duration	13 days 1 hour

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 $153 = 1^3 + 5^3 + 3^3$.

NAME: (optional)

```

1 #include<stdio.h>
2 int main()
3 {
4     int n,v,p3,c,in,i,i1,i2,t,ti;
5     scanf("%d",&t);
6     for(ti=0;ti<t;ti++){
7         v=0;
8         scanf("%d",&n);
9         printf("Case # %d\n",ti+1);
10        for(i=0;i<n;i++){
11            c=0;
12            if(i>0){
13                for(i1=0;i1<i;i1++)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)printf("0");
28            }printf("\n");
29        }
30    }
31 }

```

	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	

REC-CIS

Explanation:

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

Example 2:

Input:

123

Output:

false

Explanation:

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

Example 3:

Input:

1634

Output:

true

REC-CIS

```
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;
30 }
```

	Input	Expected	Got	
<input type="checkbox"/>	153	true	true	<input type="checkbox"/>
<input type="checkbox"/>	123	false	false	<input type="checkbox"/>

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 n, rev, sum, i=0;
```

Input:

1634

Output:

true

Note:

$1 \leq N \leq 10^8$

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     {
```

REC-CIS

Correct
Marked out of 5.00
[Flag question](#)

Sample Input 2 789 Sample Output 2 66066

Answer: (penalty regime: 0 %)

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

	Input	Expected	Got	
<input type="checkbox"/>	32	55	55	<input type="checkbox"/>
<input type="checkbox"/>	789	66066	66066	<input type="checkbox"/>

Passed all tests! ☐

Question 3
Correct

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.

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.

Sample Input 2:

34

Sample Output 2:

33344

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** Saturday, 7 December 2024, 4:11 PM**Duration** 16 days 1 hour**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.

Input:

2

3

5

Output:

WBW

REC-CIS

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;
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 }
28
```

	Input	Expected	Got	
<input type="checkbox"/>	34	33344	33344	<input type="checkbox"/>

Passed all tests! ☐

	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

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

REC-CIS

Output:

WBW

BWB

WBW

WBWBW

BWBWB

WBWBW

BWBWB

WBWBW

Answer: (penalty regime: 0 %)

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