



REC-CIS

GE23131-Programming Using C-2024

Navigation

Dashboard

[Site home](#)[Site pages](#)

My courses

GE23131-PUC-2024

[Participants](#) [Competencies](#) [Grades](#)[General](#)[Skill Test-01-MCQ & Coding](#)[Lecture Notes](#)[Week-01-Overview of C, Constants, Variables and Da...](#)[Assessment-01-Overview of C, Constants, Variables ...](#)[Week-02-Operators and Expressions, Managing Input ...](#)[Assessment-02- Operators and Expressions, Managing](#)

Attempts allowed: 3

This quiz has been configured so that students may only attempt it using the Safe Exam Browser.

Time limit: 2 hours

Grading method: Highest grade

Your attempts

Attempt 3

Status	Finished
Started	Sunday, 12 January 2025, 1:22 PM
Completed	Sunday, 12 January 2025, 2:43 PM
Duration	1 hour 20 mins

[Review](#)

Attempt 2

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Tuesday, 17 December 2024, 12:49 PM
Duration	6 days 4 hours

[Review](#)

Attempt 1

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Thursday, 12 December 2024, 10:24 AM
Duration	11 days 7 hours



Type here to search



Rain tomorrow



03:51

12-01-2025

GE23131-Programming Using C-2024

Quiz navigation

1 2 3

Show one page at a time

Finish review

Status Finished
Started Sunday, 12 January 2025, 1:22 PM
Completed Sunday, 12 January 2025, 2:43 PM
Duration 1 hour 20 mins

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 W for white spaces and B for black spaces.

Input:

2

2

WBW

BWB

WBW

WBWBW

BWBWB

WBWBW

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int T, size;
5     scanf("%d",&T);
6     while(T--)
7     {
8         scanf("%d",&size);
9
10        for(int i=0;i<size;i++)
11        {
12            for(int j=0;j<size;j++)
13            {
14                if((i+j)%2==0)
15                {
16                    printf("W");
17                }
18                else
19                {
```

```
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
```

```
{         printf("W");
}
else
{
    printf("B");
}
printf("\n");
}
return 0;
```

	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

Let's print a chessboard!



Type here to search



Rain tomorrow



ENG

03:51
12-01-2025

			WBWBW	WBWBW	WBWBW	
			BWBWB	BWBWB	BWBWB	
			WBWBW	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

2 B



Type here to search



Rain tomorrow



ENG

03:51

12-01-2025

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            {
16                c=(i2%2==o)? '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
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

REC-CIS

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



Rain tomorrow

03:51

12-01-2025

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 #%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++)
```

```
28         printf("%d",p3++);
29         if (i2!=n-1) printf("0");
30     }printf("\n");
31 }
32 }
33 }
34 }
```

	Input	Expected	Got	
✓	3	Case #1 10203010011012	Case #1 10203010011012	✓
	4	**4050809	**4050809	
	5	****607	****607	
		Case #2 1020304017018019020 **50607014015016 ****809012013 *****10011	Case #2 1020304017018019020 **50607014015016 ****809012013 *****10011	
		Case #3 102030405026027028029030 **6070809022023024025 ****10011012019020021 *****13014017018 *****15016	Case #3 102030405026027028029030 **6070809022023024025 ****10011012019020021 *****13014017018 *****15016	

Passed all tests! ✓

Finish review



REC-CIS

GE23131-Programming Using C-2024

Navigation

Dashboard

[Site home](#)[Site pages](#)

My courses

GE23131-PUC-2024

[Participants](#) [Competencies](#) [Grades](#)[General](#)[Skill Test-01-MCQ & Coding](#)[Lecture Notes](#)[Week-01-Overview of C, Constants, Variables and Da...](#)[Assessment-01-Overview of C, Constants, Variables ...](#)[Week-02-Operators and Expressions, Managing Input ...](#)[Assessment-02- Operators and Expressions, Managing](#)

Attempts allowed: 3

This quiz has been configured so that students may only attempt it using the Safe Exam Browser.

Time limit: 2 hours

Grading method: Highest grade

Your attempts

Attempt 1

Status Finished

Started Monday, 23 December 2024, 5:33 PM

Completed Tuesday, 17 December 2024, 10:47 AM

Duration 6 days 6 hours

[Review](#)

The Safe Exam Browser keys could not be validated. Check that you're using Safe Exam Browser with the correct configuration file.

[Download Safe Exam Browser](#)[Launch Safe Exam Browser](#)[Download configuration](#)[Type here to search](#)

Rain tomorrow



ENG

03:51

12-01-2025

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, 17 December 2024, 10:47 AM
Duration	6 days 6 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:



Type here to search



Rain tomorrow



ENG

03:51
12-01-2025

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



Rain tomorrow



03:51

12-01-2025



Type here to search



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



Rain tomorrow

03:51
12-01-2025

Type here to search



REC-CIS

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;
```

REC-CIS

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;
```

```
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	
✓	153	true	true	✓
✓	123	false	false	✓

Passed all tests! ✓

Question 2
Correct
Marked out of

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

```
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	
✓	153	true	true	✓
✓	123	false	false	✓

Passed all tests! ✓

Question 2
Correct
Marked out of

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

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

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



Rain tomorrow



ENG

03:51

12-01-2025



Type here to search



Passed all tests! ✓

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:



Type here to search



Rain tomorrow



ENG

03:51
12-01-2025

Please enter lucky numbers and 0, 1, 33, 34, and the unlucky number is 333.

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;
9         while(nt!=0)
10        {
11             co=0;
12             if(nt%10!=3&&nt%10!=4)
13             {
14                 co=1;
15                 break;
```

```
/ {  
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 }
```

	Input	Expected	Got	
✓	34	33344	33344	✓

Passed all tests! ✓

Finish review

Assessment-05-Decision Making and Looping - while and do...while

Dashboard / My courses / GE23131-PUC-2024 / Assessment-05-Decision Making and Looping - while ...

Navigation

Dashboard

 Site home

 Site pages

My courses

 GE23131-PUC-2024

 Participants

 Competencies

 Grades

 General

 Skill Test-01-MCQ &

Coding

 Lecture Notes

 Week-01-Overview of C, Constants, Variables and Da...

 Assessment-01-Overview of C, Constants, Variables ...

 Week-02-Operators and Expressions, Managing Input ...

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

 Week-06-Decision Making and Looping - for

 Coding

 Done

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

Jump to...

 Week-06-Decision Making and Looping - for

Week-06-One-Dimensional Arrays

[Dashboard](#) / [My courses](#) / [GE23131-PUC-2024](#) / [Week-06-One-Dimensional Arrays](#)

Navigation

Dashboard

 Site home

 Site pages

My courses

 GE23131-PUC-2024

 Participants

 Competencies

 Grades

 General

 Skill Test-01-MCQ &
Coding

 Lecture Notes

 Week-01-Overview of C,
Constants, Variables and
Da...

 Assessment-01-Overview
of C, Constants, Variables ...

 Week-02-Operators and
Expressions, Managing

 [Assessment-07-Nested Loops - while and for, Jumps in Loops](#)

 [Week-06-01-One-Dimensional Arrays](#)

 [Week-06-01-Practice Session-Coding](#)

 Done

 [Assessment-07-Nested Loops - while
and for, Jumps in Loops](#)

Jump to...

 [Week-06-01-One-
Dimensional Arrays](#)

REC-CIS

GE23131-Programming Using C-2024

Navigation

Dashboard

[Site home](#)[Site pages](#)

My courses

GE23131-PUC-2024

[Participants](#) [Competencies](#) [Grades](#)[General](#)[Skill Test-01-MCQ & Coding](#)[Lecture Notes](#)[Week-01-Overview of C, Constants, Variables and Da...](#)[Assessment-01-Overview of C, Constants, Variables ...](#)[Week-02-Operators and Expressions, Managing Input ...](#)[Assessment-02- Operators and Expressions, Managing](#)

Attempts allowed: 3

This quiz has been configured so that students may only attempt it using the Safe Exam Browser.

Time limit: 2 hours

Grading method: Highest grade

Your attempts

Attempt 1

Status Finished

Started Monday, 23 December 2024, 5:33 PM

Completed Thursday, 19 December 2024, 8:42 AM

Duration 4 days 8 hours

[Review](#)

The Safe Exam Browser keys could not be validated. Check that you're using Safe Exam Browser with the correct configuration file.

[Download Safe Exam Browser](#)[Launch Safe Exam Browser](#)[Download configuration](#) Type here to search

USD/INR +0.32%



03:53

12-01-2025

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 Thursday, 19 December 2024, 8:42 AM
Duration 4 days 8 hours

Question 1

Correct

Marked out of
3.00

Flag question

Given an array A of sorted integers and another non negative integer k, find if there exists 2 indices i and j such that $A[i] - A[j] = k$, $i \neq j$.

Input Format

1. First line is number of test cases T. Following T lines contain:
2. N, followed by N integers of the array
3. The non-negative integer k

Output format

Print 1 if such a pair exists and 0 if it doesn't.

Example

Input:



USD/INR +0.32%

03:53
12-01-2025



REC-CIS

Output format

Print 1 if such a pair exists and 0 if it doesn't.

Example**Input:**

1

3 1 3 5

4

Output:

1

Input:

1

3 1 3 5

99

Output:

0

Answer: (penalty regime: 0 %)

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

	Input	Expected	Got	
✓	1 3 1 3 5 4	1	1	✓
✓	1 3 1 3 5 99	0	0	✓

Passed all tests! ✓

Question 2

Correct

Marked out of
5.00[Flag question](#)

Sam loves chocolates and starts buying them on the 1st day of the year. Each day of the year, x , is numbered from 1 to Y . On days when x is odd, Sam will buy x chocolates; on days when x is even, Sam will not purchase any chocolates.

Complete the code in the editor so that for each day N_i (where $1 \leq x \leq N \leq Y$) in array arr , the number of chocolates Sam purchased (during days 1 through N) is printed on a new line. This is a function-only challenge, so input is handled for you by the locked stub code in the editor.

Input Format

The program takes an array of integers as a parameter.

The locked code in the editor handles reading the following input from `stdin`, assembling it into an array of integers (`arr`), and calling `calculate(arr)`.

The first line of input contains an integer, T (the number of test cases). Each line i of the T subsequent lines describes the i th test case as an integer, N_i (the number of days).

REC-CIS

`calculate(arr).`

The first line of input contains an integer, T (the number of test cases). Each line i of the T subsequent lines describes the i th test case as an integer, N_i (the number of days).

Constraints

$$1 \leq T \leq 2 \times 10^5$$

$$1 \leq N \leq 2 \times 10^6$$

$$1 \leq x \leq N \leq Y$$

Output Format

For each test case, T_i in arr , your `calculate` method should print the total number of chocolates Sam purchased by day N_i on a new line.

Sample Input 0

```
3
1
2
3
```

Sample Output 0

Test Case 0: N = 1

Sam buys 1 chocolate on day 1, giving us a total of 1 chocolate. Thus, we print 1 on a new line.

Test Case 1: N = 2

Sam buys 1 chocolate on day 1 and 0 on day 2. This gives us a total of 1 chocolate. Thus, we print 1 on a new line.

Test Case 2: N = 3

Sam buys 1 chocolate on day 1, 0 on day 2, and 3 on day 3. This gives us a total of 4 chocolates. Thus, we print 4 on a new line.

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int t;
5     scanf("%d",&t);
6     while(t--)
7     {
8         int n,c=0;
9         scanf("%d",&n);
10        for(int i=0;i<=n;i++)
11        {
12            if(i%2!=0)
13                c=c+i;
14        }
15        printf("%d\n",c);
16    }
17 }
```

Week-06-01-Practice Session-Coding: Attempt review | REC-CIS - Google Chrome

Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=126895&cmid=178

REC-CIS

54	400	400
40	25	25
9	1521	1521
77	25	25
9	49	49
13	2401	2401
98		

Passed all tests! ✓

Question 3

Correct

Marked out of
7.00

[Flag question](#)

The number of goals achieved by two football teams in matches in a league is given in the form of two lists. Consider:

- Football team A, has played three matches, and has scored { 1 , 2 , 3 } goals in each match respectively.
- Football team B, has played two matches, and has scored { 2, 4 } goals in each match respectively.
- Your task is to compute, for each match of team B, the total number of matches of team A, where team A has scored less than or equal to the number of goals scored by team B in that match.
- In the above case:
- For 2 goals scored by team B in its first match, team A has 2 matches with scores 1 and 2.
- For 4 goals scored by team B in its second match, team A has 3 matches with scores 1, 2 and 3.

Hence, the answer: {2, 3}.

Complete the code in the editor below. The program must return an array of m positive integers, one for each maxes[i] representing the total number of elements nums[j] satisfying $\text{nums}[j] \leq \text{maxes}[i]$ where $0 \leq j < n$ and $0 \leq i < m$, in the given order.

Complete the code in the editor below. The program must return an array of m positive integers, one for each $\text{maxes}[i]$ representing the total number of elements $\text{nums}[j]$ satisfying $\text{nums}[j] \leq \text{maxes}[i]$ where $0 \leq j < n$ and $0 \leq i < m$, in the given order.

It has the following:

$\text{nums}[\text{nums}[0], \dots, \text{nums}[n-1]]$: first array of positive integers

$\text{maxes}[\text{maxes}[0], \dots, \text{maxes}[n-1]]$: second array of positive integers

Constraints

- $2 \leq n, m \leq 105$
- $1 \leq \text{nums}[j] \leq 109$, where $0 \leq j < n$.
- $1 \leq \text{maxes}[i] \leq 109$, where $0 \leq i < m$.

Input Format For Custom Testing

Input from `stdin` will be processed as follows and passed to the function.

The first line contains an integer n , the number of elements in nums .

The next n lines each contain an integer describing $\text{nums}[j]$ where $0 \leq j < n$.

The next line contains an integer m , the number of elements in maxes .

The next m lines each contain an integer describing $\text{maxes}[i]$ where $0 \leq i < m$.

REC-CIS

The next m lines each contain an integer describing maxes[i] where $0 \leq i < m$.

Sample Case 0

Sample Input 0

```
4  
1  
4  
2  
4  
2  
3  
5
```

Sample Output 0

```
2  
4
```

Explanation 0

Explanation 0

We are given $n = 4$, $\text{nums} = [1, 4, 2, 4]$, $m = 2$, and $\text{maxes} = [3, 5]$.

1. For $\text{maxes}[0] = 3$, we have 2 elements in nums ($\text{nums}[0] = 1$ and $\text{nums}[2] = 2$) that are $\leq \text{maxes}[0]$.
2. For $\text{maxes}[1] = 5$, we have 4 elements in nums ($\text{nums}[0] = 1$, $\text{nums}[1] = 4$, $\text{nums}[2] = 2$, and $\text{nums}[3] = 4$) that are $\leq \text{maxes}[1]$.

Thus, the function returns the array $[2, 4]$ as the answer.

Sample Case 1

Sample Input 1

5
2
10
5
4
8
4
3
1
7

4

Explanation 1

We are given, $n = 5$, $\text{nums} = [2, 10, 5, 4, 8]$, $m = 4$, and $\text{maxes} = [3, 1, 7, 8]$.

1. For $\text{maxes}[0] = 3$, we have 1 element in nums ($\text{nums}[0] = 2$) that is $\leq \text{maxes}[0]$.
2. For $\text{maxes}[1] = 1$, there are 0 elements in nums that are $\leq \text{maxes}[1]$.
3. For $\text{maxes}[2] = 7$, we have 3 elements in nums ($\text{nums}[0] = 2$, $\text{nums}[2] = 5$, and $\text{nums}[3] = 4$) that are $\leq \text{maxes}[2]$.
4. For $\text{maxes}[3] = 8$, we have 4 elements in nums ($\text{nums}[0] = 2$, $\text{nums}[2] = 5$, $\text{nums}[3] = 4$, and $\text{nums}[4] = 8$) that are $\leq \text{maxes}[3]$.

Thus, the function returns the array $[1, 0, 3, 4]$ as the answer.

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int s1,s2,ans;
5     scanf("%d",&s1);
6     int ta[s1];
7     for(int i=0;i<s1;i++)
8         scanf("%d",&ta[i]);
9     scanf("%d",&s2);
10    int tb[s2];
11    for(int i=0;i<s2;i++)
12        scanf("%d",&tb[i]);
13    for(int j=0;j<s2;j++)
14    {
15        ans=0;
16        for(int i=0;i<s1;i++)
17        {
```

REC-CIS

```
1/1 {  
18     if(tb[j]>=ta[i])  
19         ans++;  
20     }  
21     printf("%d\n",ans);  
22 }  
23 }
```

	Input	Expected	Got	
✓	4	2	2	✓
	1	4	4	
	4			
	2			
	4			
	2			
	3			
	5			
✓	5	1	1	✓
	2	0	0	
	10	3	3	
	5	4	4	
	4			
	8			
	4			
	3			
	1			
	7			
	8			

Passed all tests! ✓

23 | }

	Input	Expected	Got	
✓	4	2	2	✓
	1	4	4	
	4			
	2			
	4			
	2			
	3			
	5			
✓	5	1	1	✓
	2	0	0	
	10	3	3	
	5	4	4	
	4			
	8			
	4			
	3			
	1			
	7			
	8			

Passed all tests! ✓

Finish review

23 | }

	Input	Expected	Got	
✓	4	2	2	✓
	1	4	4	
	4			
	2			
	4			
	2			
	3			
	5			
✓	5	1	1	✓
	2	0	0	
	10	3	3	
	5	4	4	
	4			
	8			
	4			
	3			
	1			
	7			
	8			

Passed all tests! ✓

Finish review

23 | }

	Input	Expected	Got	
✓	4	2	2	✓
	1	4	4	
	4			
	2			
	4			
	2			
	3			
	5			
✓	5	1	1	✓
	2	0	0	
	10	3	3	
	5	4	4	
	4			
	8			
	4			
	3			
	1			
	7			
	8			

Passed all tests! ✓

Finish review

23 | }

	Input	Expected	Got	
✓	4	2	2	✓
	1	4	4	
	4			
	2			
	4			
	2			
	3			
	5			
✓	5	1	1	✓
	2	0	0	
	10	3	3	
	5	4	4	
	4			
	8			
	4			
	3			
	1			
	7			
	8			

Passed all tests! ✓

Finish review