

**Question 2**

Correct

Marked out of  
5.00[Flag question](#)

The Chinese zodiac assigns animals to years in a 12-year cycle. One 12-year cycle is shown in the table below. The pattern repeats from there, with 2012 being another year of the Dragon, and 1999 being another year of the Hare.

Year	Animal
2000	Dragon
2001	Snake
2002	Horse
2003	Sheep
2004	Monkey
2005	Rooster
2006	Dog
2007	Pig
2008	Rat
2009	Ox
2010	Tiger
2011	Hare

Write a program that reads a year from the user and displays the animal associated with that year. Your program should work correctly for any year greater than or equal to zero, not just the ones listed in the table.

Sample Input 1

2004

Sample Output 1

Monkey

Sample Input 2

2010

Sample Output 2

Tiger

**Answer:** (penalty regime: 0 %)

```

1 #include<stdio.h>
2 int main(){
3     int a;
4     scanf("%d",&a);
5     if(a%12==8){
6         printf("Dragon");
7     }else if(a%12==9){
8         printf("Snake");
9     }else if(a%12==10){
10        printf("Horse");
11    }else if(a%12==11){
12        printf("Sheep");
13    }else if(a%12==0){
14        printf("Monkey");
15    }else if(a%12==1){
16        printf("Rooster");
17    }else if(a%12==2){
18        printf("Dog");
19    }else if(a%12==3){
20        printf("Pig");
21    }else if(a%12==4){
22        printf("Rat");
23    }else if(a%12==5){
24        printf("Ox");
25    }else if(a%12==6){
26        printf("Tiger");
27    }else if(a%12==7){
28        printf("Hare");
29    }
30 }
```

	Input	Expected	Got	
✓	2004	Monkey	Monkey	✓
✓	2010	Tiger	Tiger	✓

Passed all tests! ✓



## Week-03-02-Practice Ses...



rajalakshmicolleges.org

REC-CIS

Finish review

Question 1

Correct

Marked out of  
3.00[Flag question](#)

Write a program that determines the name of a shape from its number of sides. Read the number of sides from the user and then report the appropriate name as part of a meaningful message. Your program should support shapes with anywhere from 3 up to (and including) 10 sides. If a number of sides outside of this range is entered then your program should display an appropriate error message.

Sample Input 1

3

Sample Output 1

Triangle

Sample Input 2

7

Sample Output 2

Heptagon

Sample Input 3

11

Sample Output 3

The number of sides is not supported.

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main(){
3     int s;
4     scanf("%d",&s);
5     if(s==3){
6         printf("Triangle");
7     }else if(s==4){
8         printf("Square");
9     }else if(s==5){
10        printf("Pentagon");
11    }else if(s==6){
12        printf("Hexagon");
13    }else if(s==7){
14        printf("Heptagon");
15    }else if(s==8){
16        printf("Octagon");
17    }else if(s==9){
18        printf("Nonagon");
19    }else if(s==10){
20        printf("Decagon");
21    }else{
22        printf("The number of sides is no
23    }
```

	Input	Expected
✓	3	Triangle
✓	7	Heptagon
✓	11	The number of sides is not supported.

Passed all tests! ✓



Sample Case 0: n = 3

n is odd and odd numbers are weird, so we print **Weird**.

Sample Case 1: n = 24

n > 20 and n is even, so it isn't weird. Thus, we print **Not Weird**.

Answer: (penalty regime: 0 %)

```

1 #include<stdio.h>
2 int main(){
3     int a;
4     scanf("%d",&a);
5     if (a%2!=0){
6         printf("Weird");
7     }else if(a%2==0 && 2<=a && a<=6){
8         printf("Not Weird");
9     }else if(a%2==0 && 6<=a && a<=20){
10        printf("Weird");
11    }else if(a%2==0 && a>20){
12        printf("Not Weird");
13    }
14 }
15 }
```

	Input	Expected	Got	
✓	3	Weird	Weird	✓
✓	24	Not Weird	Not Weird	✓

Passed all tests! ✓

## Question 3

Correct

Marked out of 7.00

[Flag question](#)

Three numbers form a Pythagorean triple if the sum of squares of two numbers is equal to the square of the third. For example, 3, 5 and 4 form a Pythagorean triple, since  $3^2 + 4^2 = 5^2$ . You are given three integers, a, b, and c. They need not be given in increasing order. If they form a Pythagorean triple, then print "yes", otherwise, print "no". Please note that the output message is in small letters.

Sample Input 1 3 5 4 Sample Output 1 yes Sample Input 2 5 8 2 Sample Output 2 no

Answer: (penalty regime: 0 %)

```

1 #include<stdio.h>
2 int main() {
3     int a, b, c;
4     scanf("%d %d %d",&a,&b,&c);
5     if (a>b && a>c){
6         if(a*a == b*b + c*c){
7             printf("yes");
8         }else{
9             printf("no");
10        }
11    }else if (b>a && b>c){
12        if(b*b == a*a + c*c){
13            printf("yes");
14        }else{
15            printf("no");
16        }
17    }else {
18        if(c*c == a*a + b*b){
19            printf("yes");
20        }else {
21            printf("no");
22        }
23    }
24 }
```

	Input	Expected	Got	
✓	3 5 4	yes	yes	✓
✓	5 8 2	no	no	✓

Passed all tests! ✓

[Finish review](#)

**Question 2**

Correct

Marked out of  
5.00[Flag question](#)**Objective**

In this challenge, we're getting started with conditional statements.

**Task**

Given an integer, *n*, perform the following conditional actions:

- If *n* is odd, print **Weird**
- If *n* is even and in the inclusive range of 2 to 5, print **Not Weird**
- If *n* is even and in the inclusive range of 6 to 20, print **Weird**
- If *n* is even and greater than 20, print **Not Weird**

Complete the stub code provided in your editor to print whether or not *n* is weird.

**Input Format**

A single line containing a positive integer, *n*.

**Constraints**

- $1 \leq n \leq 100$

**Output Format**

Print **Weird** if the number is weird; otherwise, print **Not Weird**.

**Sample Input 0**

3

**Sample Output 0**

Weird

**Sample Input 1**

24

**Sample Output 1**

Not Weird

**Explanation****Sample Case 0: *n* = 3**

*n* is odd and odd numbers are weird, so we print **Weird**.

**Sample Case 1: *n* = 24**

*n* > 20 and *n* is even, so it isn't weird. Thus, we print **Not Weird**.

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main(){
3     int a;
4     scanf("%d",&a);
5     if (a%2!=0){
6         printf("Weird");
7     }else if(a%2==0 && 2<=a && a<=6){
8         printf("Not Weird");
9     }else if(a%2==0 && 6<=a && a<=20){
10        printf("Weird");
11    }else if(a%2==0 && a>20){
12        printf("Not Weird");
13    }
14    return 0;
15 }
```

[Flag question](#)

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

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

Note:

$1 \leq N \leq 10^8$

**Answer:** (penalty regime: 0 %)

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

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

Passed all tests! ✓



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 Friday, 25 October 2024, 12:44 PM

Duration 59 days 4 hours

## Question 1

Correct

Marked out of  
3.00

Flag question

Write a program to read two integer values and print true if both the numbers end with the same digit, otherwise print false. Example: If 698 and 768 are given, program should print true as they both end with 8. Sample Input 1 25 53 Sample Output 1 false Sample Input 2 27 77 Sample Output 2 true

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main() {
3     int a, b, resa, resb;
4     scanf("%d %d", &a, &b);
5     resa = a%10;
6     resb = b%10;
7     if(resa == resb){
8         printf("true");
9     }else{
10        printf("false");
11    }
12    return 0;
13 }
```

	Input	Expected	Got	
✓	25 53	false	false	✓
✓	27 77	true	true	✓

Passed all tests! ✓

## Question 2

Correct

Marked out of  
5.00

Flag question

## Objective

In this challenge, we're getting started with conditional statements.

## Task

Given an integer,  $n$ , perform the following conditional actions:

- If  $n$  is odd, print Weird
- If  $n$  is even and in the inclusive range of 2 to 5, print **Not Weird**
- If  $n$  is even and in the inclusive range of 6 to 20, print **Weird**
- If  $n$  is even and greater than 20, print **Not Weird**

Complete the stub code provided in your editor to print whether or not  $n$  is weird.

## Input Format

A single line containing a positive integer,  $n$ .



## 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, 2:31 PM  
Duration: 13 days 3 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  $1^3 + 2^3 + 3^3 = 36 \neq 123$ .

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 int main(){
4     int n, a, p=0, t, sum=0, x;
5     scanf("%d", &n);
6     t=n;
7     a=n;
8     while(t!=0){
9         t=t/10;
10        p++;
11    }while(n!=0){
12        x=n%10;
13        sum=sum+pow(x,p);
14        n=n/10;
15    }
16    if(sum==a){
17        printf("true");
18    }else{
19    }
```

Answer: (penalty regime: 0 %)

```

1 #include<stdio.h>
2 int main() {
3     int N;
4     scanf("%d",&N);
5     printf("%d",N*(N-1)/2);
6     return 0;
7 }
```

	Input	Expected	Got	
✓	1	0	0	✓
✓	2	1	1	✓

Passed all tests! ✓

## Question 3

Correct

Marked out of  
7.00[Flag question](#)

In our school days, all of us have enjoyed the Games period. Raghab loves to play cricket and is Captain of his team. He always wanted to win all cricket matches. But only one last Games period is left in school now. After that he will pass out from school. So, this match is very important to him. He does not want to lose it. So he has done a lot of planning to make sure his teams wins. He is worried about only one opponent - Jatin, who is very good batsman. Raghab has figured out 3 types of bowling techniques, that could be most beneficial for dismissing Jatin. He has given points to each of the 3 techniques. You need to tell him which is the maximum point value, so that Raghab can select best technique. 3 numbers are given in input. Output the maximum of these numbers.

Input:

Three space separated integers.

Output:

Maximum integer value

SAMPLE INPUT

8 6 1

SAMPLE OUTPUT

8

Explanation Out of given numbers, 8 is maximum.

Answer: (penalty regime: 0 %)

```

1 #include<stdio.h>
2 int main(){
3     int a,b,c;
4     scanf("%d %d %d",&a,&b,&c);
5     if(a>b){
6         if(a>c){
7             printf("%d",a);
8         }else{
9             printf("%d",c);
10        }
11    }else{
12        if(b>c){
13            printf("%d",b);
14        }else{
15            printf("%d",c);
16        }
17    }
18    return 0;
19 }
```

	Input	Expected	Got	
✓	81 26 15	81	81	✓

Passed all tests! ✓

[Finish review](#)

```

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 t, n, x, y, i, z=1, ans, c;
4     scanf("%d",&t);
5     while(z<=t){
6         scanf("%d",&n);
7         printf("Case #%d\n", z);
8         y = 1;
9         i = 1;
10        c = 0;
11        while(y<=n){
12            x = 1;
13            ans = (n*n);
14            ans = ans - c;
15            while (x<= 2^n){
16                if ( x<= n){
17                    if (x<y)
18                        printf("****");
19                    else if(x <= n){
20                        printf("%d",i*10);
21                        i++;
22                    }else{
23                        if((x+y)==(2^n)+1){
24                            printf("%d",ans
25                            ans++;
26                            c++;
27                        }else if((x+y)<=(2^n)+1){
28                            printf("%d", (ans + y)
29                            ans++;
30                            c++;
31                        }
32                    }
33                    x++;
34                }
35                y++;
36            }
37        }
38    }
39 }
40
41

```

	Input	Expected	Got
✓	3	Case #1 10203010011012 **4050809 ****607	Case #1 102030100110 **4050809 ****607
	3	Case #2 1020304017018019020 **50607014015016 ****809012013 *****10011	Case #2 102030401701 **5060701401 ****80901201 *****10011
	4	Case #3 102030405026027028029030 **6070809022023024025 ****10011012019020021 *****13014017018 *****15016	Case #3 102030405026 **6070809022 ****10011012 *****130140 *****1501
	5		

Passed all tests! ✓

Finish review

**Question 2**

Correct

Marked out of  
5.00[Flag question](#)

Before the outbreak of corona virus to the world, a meeting happened in a room in Wuhan. A person who attended that meeting had COVID-19 and no one in the room knew about it! So everyone started shaking hands with everyone else in the room as a gesture of respect and after meeting unfortunately everyone got infected! Given the fact that any two persons shake hand exactly once, Can you tell the total count of handshakes happened in that meeting? Say no to shakehands. Regularly wash your hands. Stay Safe.

**Input Format**

Read an integer N, the total number of people attended that meeting.

**Output Format**

Print the number of handshakes.

**Constraints**

$0 < N < 106$

**SAMPLE INPUT 1**

1

**SAMPLE OUTPUT**

0

**SAMPLE INPUT 2**

2

**SAMPLE OUTPUT 2**

1

Explanation Case 1: The lonely board member shakes no hands, hence 0. Case 2: There are 2 board members, 1 handshake takes place.

**Answer:** (penalty regime: 0 %)

```

1 #include<stdio.h>
2 int main() {
3     int N;
4     scanf("%d",&N);
5     printf("%d",N*(N-1)/2);
6     return 0;
7 }
```

	Input	Expected	Got	
✓	1	0	0	✓
✓	2	1	1	✓

Passed all tests! ✓

**Question 3**

Correct

Marked out of  
7.00[Flag question](#)

In our school days, all of us have enjoyed the Games period. Raghav loves to play cricket and is Captain of his team. He always wanted to win all cricket matches. But only one last Games period is left in school now. After that he will pass out from school. So, this match is very important to him. He does not want to lose it. So he has done a lot of planning to make sure his team wins. He is worried about only one opponent - Jatin, who is very good batsman. Raghav has figured out 3 types of bowling techniques, that could be most beneficial for dismissing Jatin. He has given points to each of the 3 techniques. You need to tell him which is the maximum point value, so that Raghav can select best technique. 3 numbers are given in input. Output the maximum of these numbers.

**Input:**

Three space separated integers.

**Output:**

Maximum integer value

**SAMPLE INPUT**

8 6 1

**SAMPLE OUTPUT**

8

**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 &lt;= N &lt;= 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 t, n, x, y, i, z=1, ans, c;
4     scanf("%d", &t);
5     while(z<=t){
6         scanf("%d", &n);
7         printf("Case # %d\n", z);
8         y = 1;
9         i = 1;
10        c = 0;
11        while(y<=n){
12            x = 1;
13            ans = (n*n);
14            ans = ans - c;
15            c = 2 * c + 1;
16            y++;
17        }
18    }
19}
```



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 Tuesday, 22 October 2024, 2:53 PM

Duration 62 days 2 hours

## Question 1

Correct

Marked out of 3.00

Flag question

Goki recently had a breakup, so he wants to have some more friends in his life. Goki has N people who he can be friends with, so he decides to choose among them according to their skills set  $Y_i (1 \leq i \leq n)$ . He wants atleast X skills in his friends. Help Goki find his friends.

## INPUT

First line contains a single integer X - denoting the minimum skill required to be Goki's friend. Next line contains one integer Y - denoting the skill of the person

## OUTPUT

Print if he can be friend with Goki. 'YES' (without quotes) if he can be friends with Goki else 'NO' (without quotes).

## CONSTRAINTS

 $1 \leq N \leq 1000000$  $1 \leq X, Y \leq 1000000$ 

## SAMPLE INPUT 1

100 110

## SAMPLE OUTPUT 1

YES

## SAMPLE INPUT 2

100 90

## SAMPLE OUTPUT 2

NO

## Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main(){
3     int skillsreq;
4     int skill;
5     scanf("%d",&skill);
6     scanf("%d",&skillsreq);
7     if(skillsreq>=skill){
8         printf("YES");
9     }else{
10         printf("NO");
11     }
12     return 0;
13 }
```

	Input	Expected	Got	
✓	100 110	YES	YES	✓
✓	100 90	NO	NO	✓

Passed all tests! ✓

## Question 2

Before the outbreak of corona virus to the world, a meeting



```

4 int b;
5 scanf("%d",&a);
6 scanf("%d",&b);
7 printf("%d\n",a+b);
8 printf("%d\n",a-b);
9 printf("%d\n",a*b);
10 printf("%d\n",a/b);
11 printf("%d\n",a%b);
12 return 0;
13 }

```

	Input	Expected	Got	
✓	100 6	106 94 600 16 4	106 94 600 16 4	✓

Passed all tests! ✓

**Question 3**

Correct

Marked out of  
7.00[Flag question](#)

A bakery sells loaves of bread for \$3.49 each. Day old bread is discounted by 60 percent. Write a program that begins by reading the number of loaves of day old bread being purchased from the user. Then your program should display the regular price for the bread, the discount because it is a day old, and the total price. Each of these amounts should be displayed on its own line with an appropriate label. All of the values should be displayed using two decimal places.

Input Format

Read the number of day old loaves.

Output Format

First line, print Regular price: price

Second line, print Discount: discount

Third line, print Total: total

Note: All of the values should be displayed using two decimal places.

Sample Input 1

10

Sample Output 1

Regular price: 34.90

Discount: 20.94

Total: 13.96

Answer: (penalty regime: 0 %)

```

1 #include<stdio.h>
2 int main() {
3     int amount;
4     float price;
5     scanf("%d",&amount);
6     printf("Regular price: %.2f\n",price=-
7     printf("Discount: %.2f\n",price*0.60);
8     printf("Total: %.2f\n",price*0.40);
9     return 0;
10 }
11

```

	Input	Expected	Got
✓	10	Regular price: 34.90 Discount: 20.94 Total: 13.96	Regular price: 3 Discount: 20.94 Total: 13.96

Passed all tests! ✓

[Finish review](#)

**Question 2**

Correct

Marked out of  
5.00[Flag question](#)

Create a program that reads two integers, a and b, from the user. Your program should compute and display:

- The sum of a and b
- The difference when b is subtracted from a
- The product of a and b
- The quotient when a is divided by b
- The remainder when a is divided by b

## Input Format

First line, read the first number.

Second line, read the second number.

## Output Format

First line, print the sum of a and b

Second line, print the difference when b is subtracted from a

Third line, print the product of a and b

Fourth line, print the quotient when a is divided by b

Fifth line, print the remainder when a is divided by b

## Sample

Input 1 100 6

## Sample Output

106 94 600 16 4

**Answer:** (penalty regime: 0 %)

```

1 #include<stdio.h>
2 int main() {
3     int a;
4     int b;
5     scanf("%d", &a);
6     scanf("%d", &b);
7     printf("%d\n", a+b);
8     printf("%d\n", a-b);
9     printf("%d\n", a*b);
10    printf("%d\n", a/b);
11    printf("%d\n", a%b);
12 }
13 }
```

	Input	Expected	Got	
✓	100	106	106	✓
	6	94	94	
		600	600	
		16	16	
		4	4	

Passed all tests! ✓

**Question 3**

Correct

Marked out of  
7.00[Flag question](#)

A bakery sells loaves of bread for \$3.49 each. Day old bread is discounted by 60 percent. Write a program that begins by reading the number of loaves of day old bread being purchased from the user. Then your program should display the regular price for the bread, the discount because it is a day old, and the total price. Each of these amounts should be displayed on its own line with an appropriate label. All of the values should be displayed using two decimal places.

## Input Format

Read the number of day old loaves.

## Output Format

First line, print Regular price: price

Second line, print Discount: discount

Third line, print Total: total

Note: All of the values should be displayed using two decimal places.

## Sample Input 1

10

## Sample Output 1

Regular price: 34.90

Discount: 20.94

Total: 13.96

**Answer:** (penalty regime: 0 %)

```

1 #include<stdio.h>
2 int main() {
3     int amount;
4     float price;
5     scanf("%d", &amount);
```



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 Tuesday, 22 October 2024, 2:23 PM

Duration 62 days 3 hours

## Question 1

Correct

Marked out of  
3.00[Flag question](#)

Many people think about their height in feet and inches, even in some countries that primarily use the metric system.

Write a program that reads a number of feet from the user, followed by a number of inches. Once these values are read, your program should compute and display the equivalent number of centimeters.

## Hint:

One foot is 12 inches.

One inch is 2.54 centimeters.

## Input Format

First line, read the number of feet.

Second line, read the number of inches.

## Output Format

In one line print the height in centimeters.

Note: All of the values should be displayed using two decimal places.

## Sample Input 1

5 6

## Sample Output 1

167.64

## Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main() {
3     int feet,inches;
4     scanf("%d",&feet);
5     scanf("%d",&inches);
6     printf("%.2f",feet*30.48+inches*2.54);
7 }
8 }
```

	Input	Expected	Got	
✓	5	167.64	167.64	✓
	6			

Passed all tests! ✓

## Question 2

Correct

Marked out of  
5.00[Flag question](#)

Create a program that reads two integers, a and b, from the user. Your program should compute and display:

- The sum of a and b
- The difference when b is subtracted from a
- The product of a and b
- The quotient when a is divided by b
- The remainder when a is divided by b

## Input Format

First line, read the first number.

Second line, read the second number.



3

**Explanation 0**

The following sequence of  $n = 2$  food items:

1. Item 1 has 1 macronutrients.
2.  $1 + 2 = 3$ ; observe that this is the max total, and having avoided having exactly  $k = 2$  macronutrients.

**Sample Input 1**

2  
1

**Sample Output 1**

2

**Explanation 1**

1. Cannot use item 1 because  $k = 1$  and  $\text{sum} \equiv k$  has to be avoided at any time.
2. Hence, max total is achieved by  $\text{sum} = 0 + 2 = 2$ .

**Sample Case 2****Sample Input For Custom Testing****Sample Input 2**

3  
3

**Sample Output 2**

5

**Explanation 2**

$2 + 3 = 5$ , is the best case for maximum nutrients.

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main(){
3     long int a, b, c = 0;
4     scanf("%ld %ld", &a, &b);
5     for(int i = 0; i <=a; i++){
6         c += i;
7         if(c == b){
8             c = c - 1;
9         }
10    }
11    printf("%ld", c % 1000000007);
12 }
13 }
```

	Input	Expected	Got	
✓	2 2	3	3	✓
✓	2 1	2	2	✓
✓	3 3	5	5	✓

Passed all tests! ✓

**Question 2**

Correct

Marked out of  
1.00[Flag question](#)

A nutritionist is labeling all the best power foods in the market. Every food item arranged in a single line, will have a value beginning from 1 and increasing by 1 for each, until all items have a value associated with them. An item's value is the same as the number of macronutrients it has. For example, food item with value 1 has 1 macronutrient, food item with value 2 has 2 macronutrients, and incrementing in this fashion.

The nutritionist has to recommend the best combination to patients, i.e. maximum total of macronutrients. However, the nutritionist must avoid prescribing a particular sum of macronutrients (an 'unhealthy' number), and this sum is known. The nutritionist chooses food items in the increasing order of their value. Compute the highest total of macronutrients that can be prescribed to a patient, without the sum matching the given 'unhealthy' number.

Here's an illustration:

Given 4 food items (hence value: 1,2,3 and 4), and the unhealthy sum being 6 macronutrients, on choosing items 1, 2, 3 -> the sum is 6, which matches the 'unhealthy' sum. Hence, one of the three needs to be skipped. Thus, the best combination is from among:

- $2 + 3 + 4 = 9$
- $1 + 3 + 4 = 8$
- $1 + 2 + 4 = 7$

Since  $2 + 3 + 4 = 9$ , allows for maximum number of macronutrients, 9 is the right answer.

Complete the code in the editor below. It must return an integer that represents the maximum total of macronutrients, modulo  $1000000007$  ( $10^9 + 7$ ).

It has the following:

*n*: an integer that denotes the number of food items

*k*: an integer that denotes the unhealthy number

**Constraints**

- $1 \leq n \leq 2 \times 10^9$
- $1 \leq k \leq 4 \times 10^{15}$

**Input Format For Custom Testing**

The first line contains an integer, *n*, that denotes the number of food items.

The second line contains an integer, *k*, that denotes the unhealthy number.

**Sample Input 0**

```
2
2
```

**Sample Output 0**

```
3
```

**Explanation 0**

The following sequence of *n* = 2 food items:

1. Item 1 has 1 macronutrients.
2.  $1 + 2 = 3$ ; observe that this is the max total, and having avoided having exactly *k* = 2 macronutrients.

**Sample Input 1**

```
2
1
```

## 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, 6:55 PM

Completed Sunday, 12 January 2025, 7:15 PM

Duration 19 mins 36 secs

## Question 1

Correct

Marked out of 1.00

[Flag question](#)

## Objective

In this challenge, we're going to use loops to help us do some simple math. Check out the Tutorial tab to learn more.

## Task

Given an integer,  $n$ , print its first **10** multiples. Each multiple  $n \times i$  (where  $1 \leq i \leq 10$ ) should be printed on a new line in the form:  $n \times i = \text{result}$ .

## Input Format

A single integer,  $n$ .

## Constraints

 $2 \leq n \leq 20$ 

## Output Format

Print **10** lines of output; each line  $i$  (where  $1 \leq i \leq 10$ ) contains the **result** of  $n \times i$  in the form:

 $n \times i = \text{result}$ .

## Sample Input

2

## Sample Output

 $2 \times 1 = 2$  $2 \times 2 = 4$  $2 \times 3 = 6$  $2 \times 4 = 8$  $2 \times 5 = 10$  $2 \times 6 = 12$  $2 \times 7 = 14$  $2 \times 8 = 16$  $2 \times 9 = 18$  $2 \times 10 = 20$ 

## Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2
3 int main(){
4     int n;
5     scanf("%d", &n);
6     if(n >= 2 && n <= 20){
7         for(int i = 1; i <= 10; i++){
8             printf("%d x %d = %d\n", n, i,
9         }
10    }
11    return 0;
12 }
```

	Input	Expected	Got	
✓	2	$2 \times 1 = 2$ $2 \times 2 = 4$ $2 \times 3 = 6$ $2 \times 4 = 8$ $2 \times 5 = 10$ $2 \times 6 = 12$ $2 \times 7 = 14$ $2 \times 8 = 16$ $2 \times 9 = 18$ $2 \times 10 = 20$	$2 \times 1 = 2$ $2 \times 2 = 4$ $2 \times 3 = 6$ $2 \times 4 = 8$ $2 \times 5 = 10$ $2 \times 6 = 12$ $2 \times 7 = 14$ $2 \times 8 = 16$ $2 \times 9 = 18$ $2 \times 10 = 20$	✓

Passed all tests! ✓

## Question 2

Correct

Marked out of 1.00

[Flag question](#)

A nutritionist is labeling all the best power foods in the market. Every food item arranged in a single line, will have a value beginning from 1 and increasing by 1 for each, until all items have a value associated with them. An item's value is the same as the number of macronutrients it has. For



REC-CIS

## GE23131-Programming Using C-2024

## Quiz navigation



Show one page at a time

Finish review

Status Finished

Started Sunday, 12 January 2025, 6:55 PM

Completed Sunday, 12 January 2025, 7:15 PM

Duration 19 mins 36 secs

## Question 1

Correct

Marked out of 1.00

Flag question

## Objective

In this challenge, we're going to use loops to help us do some simple math. Check out the Tutorial tab to learn more.

## Task

Given an integer,  $n$ , print its first **10** multiples. Each multiple  $n \times i$  (where  $1 \leq i \leq 10$ ) should be printed on a new line in the form:  $n \times i = \text{result}$ .

## Input Format

A single integer,  $n$ .

## Constraints

 $2 \leq n \leq 20$ 

## Output Format

Print **10** lines of output; each line  $i$  (where  $1 \leq i \leq 10$ ) contains the *result* of  $n \times i$  in the form:

 $n \times i = \text{result}$ .

## Sample Input

2

## Sample Output

 $2 \times 1 = 2$  $2 \times 2 = 4$  $2 \times 3 = 6$  $2 \times 4 = 8$  $2 \times 5 = 10$  $2 \times 6 = 12$  $2 \times 7 = 14$  $2 \times 8 = 16$  $2 \times 9 = 18$  $2 \times 10 = 20$ 

## Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2
3 int main(){
4     int n;
5     scanf("%d", &n);
6     if(n >= 2 && n <= 20){
7         for(int i = 1; i <= 10; i++){
8             printf("%d x %d = %d\n", n, i,
9         }
10    }
11 }
12 }
```

	Input	Expected	Got		
✓	2	$2 \times 1 = 2$ $2 \times 2 = 4$ $2 \times 3 = 6$ $2 \times 4 = 8$ $2 \times 5 = 10$ $2 \times 6 = 12$ $2 \times 7 = 14$	$2 \times 1 = 2$ $2 \times 2 = 4$ $2 \times 3 = 6$ $2 \times 4 = 8$ $2 \times 5 = 10$ $2 \times 6 = 12$ $2 \times 7 = 14$	✓	



✓	32	55	55	✓
✓	789	66066	66066	✓

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:

34

Sample Output 2:

33344

Answer: (penalty regime: 0 %)

```

1 #include<stdio.h>
2 int main(){
3     long int i,j;
4     int r,n,c=0,f;
5     scanf("%d",&n);
6     for(i=1;c<=n;i++){
7         f=0;
8         j=i;
9         while(j>0){
10             r=j%10;
11             if(r==3||r==4){
12                 j=j/10;
13             }else{
14                 f=1;
15                 break;
16             }
17         }
18         if(f==0){
19             c++;
20             if(c==n){
21                 break;
22             }
23         }
24     }
25     printf("%ld",i);
26 }
27 }
```

	Input	Expected	Got	
✓	34	33344	33344	✓

Passed all tests! ✓

[Finish review](#)

**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     long long int n, s, r,t,ts;
4     scanf("%lld",&n);
5     while(1){
6         r=0;
7         t=n;
8         while(n){
9             r=r*10+(n%10);
10            n=n/10;
11        }
12        s=t+r;
13        ts=s;
14        r=0;
15        while(s){
16            r=r*10+(s%10);
17            s=s/10;
18        }
19        if(ts==r){
20            break;
21        }
22        n=ts;
23    }
24    printf("%lld",ts);
25 }
```

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

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:

34

Sample Output 2:

33344

**Answer:** (penalty regime: 0 %)

```

1 #include<stdio.h>
2 int main(){
3     long int i,j;
4     int r,n,c=0,f;
```

Print int 3,  
followed by long 12345678912345,  
followed by char a,  
followed by float 334.23,  
followed by double 14049.30493.

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main() {
3     int a;
4     long int b;
5     char c;
6     float d;
7     double e;
8     scanf("%d %ld %c %f %lf",&a,&b,&c,&d,
9         &e);
10    printf("%d\n",a);
11    printf("%ld\n",b);
12    printf("%c\n",c);
13    printf("%.3fn",d);
14    printf("%.9f",e);
15 }
```

	Input	Expect
✓	3 12345678912345 a 334.23 14049.30493	3 12345 a 334.2 14049

Passed all tests! ✓

Question 3

Correct

Marked out of  
7.00

[Flag question](#)

Write a program to print the ASCII value and the two adjacent characters of the given character.

Input

E

Output

69

D F

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main() {
3     char ch;
4     scanf("%c",&ch);
5     printf("%d\n%c %c",ch,ch-1,ch+1);
6 }
7 }
```

	Input	Expected	Got	
✓	E	69 D F	69 D F	✓

Passed all tests! ✓

[Finish review](#)

**Reading**

To read a data type, use the following syntax:

```
scanf("formatSpecifier", &val)
```

For example, to read a *character* followed by a *double*:

```
char ch;
double d;
scanf("%c %lf", &ch, &d);
```

For the moment, we can ignore the spacing between format specifiers.

**Printing**

To print a data type, use the following syntax:

```
printf("formatSpecifier", val)
```

For example, to print a *character* followed by a *double*:

```
char ch = 'd';
double d = 234.432;
printf("%c %lf", ch, d);
```

**Note:** You can also use *cin* and *cout* instead of *scanf* and *printf*; however, if you are taking a million numbers as input and printing a million lines, it is faster to use *scanf* and *printf*.

**Input Format**

Input consists of the following space-separated values:  
*int*, *long*, *char*, *float*, and *double*, respectively.

**Output Format**

Print each element on a new line in the same order it was received as input. Note that the floating point value should be correct up to 3 decimal places and the double to 9 decimal places.

**Sample Input**

```
3 12345678912345 a 334.23 14049.30493
```

**Sample Output**

```
3
12345678912345
a
334.23
14049.304930000
```

**Explanation**

Print *int* 3,  
followed by *long* 12345678912345,  
followed by *char* a,  
followed by *float* 334.23,  
followed by *double* 14049.30493.

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main() {
3     int a;
4     long int b;
5     char c;
6     float d;
7     double e;
8     scanf("%d %ld %c %f %lf", &a, &b, &c, &d,
9         &e);
10    printf("%d\n", a);
11    printf("%ld\n", b);
12    printf("%c\n", c);
13    printf("%.3f\n", d);
14    printf("%.9f", e);
15 }
```

	Input	Expect
✓	3 12345678912345 a 334.23 14049.30493	3 12345 a 334.2 14049

Passed all tests! ✓

**Question 2**

Correct

Marked out of  
5.00[Flag question](#)

Some C data types, their format specifiers, and their most common bit widths are as follows:

- *Int ("%"d")*: 32 Bit integer
- *Long ("%"ld")*: 64 bit integer
- *Char ("%"c")*: Character type
- *Float ("%"f")*: 32 bit real value
- *Double ("%"lf")*: 64 bit real value

**Reading**

To read a data type, use the following syntax:

```
scanf("formatSpecifier", &val)
```

For example, to read a *character* followed by a *double*:

```
char ch;
```

```
double d;
```

```
scanf("%c %lf", &ch, &d);
```

For the moment, we can ignore the spacing between format specifiers.

**Printing**

To print a data type, use the following syntax:

```
printf("formatSpecifier", val)
```

For example, to print a *character* followed by a *double*:

```
char ch = 'd';
```

```
double d = 234.432;
```

```
printf("%c %lf", ch, d);
```

**Note:** You can also use *cin* and *cout* instead of *scanf* and *printf*; however, if you are taking a million numbers as input and printing a million lines, it is faster to use *scanf* and *printf*.

**Input Format**

Input consists of the following space-separated values:  
*int*, *long*, *char*, *float*, and *double*, respectively.

**Output Format**

Print each element on a new line in the same order it was received as input. Note that the floating point value should be correct up to 3 decimal places and the double to 9 decimal places.

**Sample Input**

```
3 12345678912345 a 334.23 14049.30493
```

**Sample Output**

```
3
12345678912345
a
334.230
14049.304930000
```

**Explanation**

Print *int* 3,

followed by *long* 12345678912345,

followed by *char* a,

followed by *float* 334.23,

followed by *double* 14049.30493.

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main() {
3     int a;
4     long int b;
5     char c;
6     float d;
7     double e;
8     scanf("%d %ld %c %f %lf", &a, &b, &c, &d,
9         &e);
10    printf("%d\n", a);
11    printf("%ld\n", b);
12    printf("%c\n", c);
13    printf("%.3fn", d);
14    printf("%.9f", e);
15 }
```

	Input	Expect
✓	3 12345678912345 a 334.23 14049.30493	3 12345678912345 a 334.230 14049.304930000

All the test marks are in integers and hence calculate the average in integer as well. That is, you need to print the integer part of the average only and neglect the decimal part.

Input format :

Line 1 : Name(Single character)

Line 2 : Marks scored in the 3 tests separated by single space.

Output format :

First line of output prints the name of the student.

Second line of the output prints the average mark.

Constraints

Marks for each student lie in the range 0 to 100 (both inclusive)

Sample Input 1 :

A  
3 4 6

Sample Output 1 :

A  
4

Sample Input 2 :

T  
7 3 8

Sample Output 2 :

T  
6

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main() {
3     char name;
4     int mark1,mark2,mark3;
5     scanf("%c",&name);
6     scanf("%d",&mark1);
7     scanf("%d",&mark2);
8     scanf("%d",&mark3);
9     printf("%c\n",name);
10    printf("%d",(mark1+mark2+mark3)/3);
11    return 0;
12 }
```

	Input	Expected	Got	
✓	A 3 4 6	A 4	A 4	✓
✓	T 7 3 8	T 6	T 6	✓
✓	R 0 100 99	R 66	R 66	✓

Passed all tests! ✓

**Question 3**

Correct

Marked out of  
7.00[Flag question](#)

A nutritionist is labeling all the best power foods in the market. Every food item arranged in a single line, will have a value beginning from 1 and increasing by 1 for each, until all items have a value associated with them. An item's value is the same as the number of macronutrients it has. For example, food item with value 1 has 1 macronutrient, food item with value 2 has 2 macronutrients, and incrementing in this fashion.

The nutritionist has to recommend the best combination to patients, i.e. maximum total of macronutrients. However, the nutritionist must avoid prescribing a particular sum of macronutrients (an 'unhealthy' number), and this sum is known. The nutritionist chooses food items in the increasing order of their value. Compute the highest total of macronutrients that can be prescribed to a patient, without the sum matching the given 'unhealthy' number.

Here's an illustration:

Given 4 food items (hence value: 1,2,3 and 4), and the unhealthy sum being 6 macronutrients, on choosing items 1, 2, 3 -> the sum is 6, which matches the 'unhealthy' sum. Hence, one of the three needs to be skipped. Thus, the best combination is from among:

- $2 + 3 + 4 = 9$
- $1 + 3 + 4 = 8$
- $1 + 2 + 4 = 7$

Since  $2 + 3 + 4 = 9$ , allows for maximum number of macronutrients, 9 is the right answer.

Complete the code in the editor below. It must return an integer that represents the maximum total of macronutrients, modulo  $1000000007$  ( $10^9 + 7$ ).

It has the following:

*n*: an integer that denotes the number of food items

*k*: an integer that denotes the unhealthy number

**Constraints**

- $1 \leq n \leq 2 \times 10^9$
- $1 \leq k \leq 4 \times 10^{15}$

**Input Format For Custom Testing**

The first line contains an integer, *n*, that denotes the number of food items.

The second line contains an integer, *k*, that denotes the unhealthy number.

**Sample Input 0**

2  
2

**Sample Output 0**

3

**Explanation 0**

The following sequence of  $n = 2$  food items:

1. Item 1 has 1 macronutrients.
2.  $1 + 2 = 3$ ; observe that this is the max total, and having avoided having exactly  $k = 2$  macronutrients.

**Sample Input 1**

2  
1



## Week-01-02-Practice Ses...



REC-CIS

Show one page at a time

Finish review

Duration 76 days 2 hours

Question 1

Correct

Marked out of  
3.00

Flag question

Write a program to input a name (as a single character) and marks of three tests as m1, m2, and m3 of a student considering all the three marks have been given in integer format.

Now, you need to calculate the average of the given marks and print it along with the name as mentioned in the output format section.

All the test marks are in integers and hence calculate the average in integer as well. That is, you need to print the integer part of the average only and neglect the decimal part.

Input format :

Line 1 : Name(Single character)

Line 2 : Marks scored in the 3 tests separated by single space.

Output format :

First line of output prints the name of the student.

Second line of the output prints the average mark.

Constraints

Marks for each student lie in the range 0 to 100 (both inclusive)

Sample Input 1 :

A  
3 4 6

Sample Output 1 :

A  
4

Sample Input 2 :

T  
7 3 8

Sample Output 2 :

T  
6

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main() {
3     char name;
4     int mark1,mark2,mark3;
5     scanf("%c",&name);
6     scanf("%d",&mark1);
7     scanf("%d",&mark2);
8     scanf("%d",&mark3);
9     printf("%c\n",name);
10    printf("%d,(mark1+mark2+mark3)/3);
11 }
12 }
```



**Question 2**

Correct

Marked out of  
5.00[Flag question](#)

Given a number N, return true if and only if it is a *confusing number*, which satisfies the following condition:

We can rotate digits by 180 degrees to form new digits. When 0, 1, 6, 8, 9 are rotated 180 degrees, they become 0, 1, 9, 8, 6 respectively. When 2, 3, 4, 5 and 7 are rotated 180 degrees, they become invalid. A *confusing number* is a number that when rotated 180 degrees becomes a **different** number with each digit valid.

**Example 1:**

6 -&gt; 9

Input: 6

Output: true

Explanation:

We get 9 after rotating 6, 9 is a valid number and  $9 \neq 6$ .**Example 2:**

89 -&gt; 68

Input: 89

Output: true

Explanation:

We get 68 after rotating 89, 86 is a valid number and  $86 \neq 89$ .**Example 3:**

11 -&gt; 11

Input: 11

Output: false

Explanation:

We get 11 after rotating 11, 11 is a valid number but the value remains the same, thus 11 is not a confusing number.

**Note:**

1.  $0 \leq N \leq 10^9$
2. After the rotation we can ignore leading zeros, for example if after rotation we have 0008 then this number is considered as just 8.

**Answer:** (penalty regime: 0 %)

```

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

	Input	Expected	Got	
✓	6	true	true	✓
✓	89	true	true	✓
✓	25	false	false	✓

Passed all tests! ✓

**Question 3**

Correct

Marked out of  
7.00[Flag question](#)

A nutritionist is labeling all the best power foods in the market. Every food item arranged in a single line, will have a value beginning from 1 and increasing by 1 for each, until all items have a value associated with them. An item's value is the same as the number of macronutrients it has. For example, food item with value 1 has 1 macronutrient, food item with value 2 has 2 macronutrients, and incrementing in this fashion.

- o Print the sum and difference of two int variable on a new line.

- o Print the sum and difference of two float variable rounded to one decimal place on a new line.

#### Input Format

The first line contains two integers.

The second line contains two floating point numbers.

#### Constraints

- $1 \leq \text{integer variables} \leq 10^4$
- $1 \leq \text{float variables} \leq 10^4$

#### Output Format

Print the sum and difference of both integers separated by a space on the first line, and the sum and difference of both float (scaled to 1 decimal place) separated by a space on the second line.

#### Sample Input

```
10 4
4.0 2.0
```

#### Sample Output

```
14 6
6.0 2.0
```

#### Explanation

When we sum the integers **10** and **4**, we get the integer **14**. When we subtract the second number **4** from the first number **10**, we get **6** as their difference.

When we sum the floating-point numbers **4.0** and **2.0**, we get **6.0**. When we subtract the second number **2.0** from the first number **4.0**, we get **2.0** as their difference.

**Answer:** (penalty regime: 0 %)

```
1 #include <stdio.h>
2
3 int main() {
4     int a,b;
5     float c,d;
6     scanf("%d",&a);
7     scanf("%d",&b);
8     scanf("%f",&c);
9     scanf("%f",&d);
10    printf("%d ",a+b);
11    printf("%d\n",a-b);
12    printf("%.1f ",c+d);
13    printf("%.1f",c-d);
14    return 0;
15 }
```

	Input	Expected	Got	
✓	10 4 4.0 2.0	14 6 6.0 2.0	14 6 6.0 2.0	✓
✓	20 8 8.0 4.0	28 12 12.0 4.0	28 12 12.0 4.0	✓

Passed all tests! ✓

Finish review

## 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, 12 November 2024, 2:00 PM**Duration** 41 days 3 hours**Question 1**

Correct

Marked out of  
3.00

Flag question

A set of N numbers (separated by one space) is passed as input to the program. The program must identify the count of numbers where the number is odd number.

## Input Format:

The first line will contain the N numbers separated by one space.

## Boundary Conditions:

3 &lt;= N &lt;= 50

The value of the numbers can be from -99999999 to 99999999

## Output Format:

The count of numbers where the numbers are odd numbers.

## Example Input / Output 1:

## Input:

5 10 15 20 25 30 35 40 45 50

## Output:

5

## Explanation:

The numbers meeting the criteria are 5, 15, 25, 35, 45.

**Answer:** (penalty regime: 0 %)

```

1 #include<stdio.h>
2 int main(){
3     int a,count = 0;
4     char ch;
5     while(ch != '\n'){
6         scanf("%d",&a);
7         scanf("%c",&ch);
8         if(a%2!=0){
9             count++;
10        }
11    }
12    printf("%d",count);
13    return 0;
14 }
```

	Input	Expected	Got
✓	5 10 15 20 25 30 35 40 45 50	5	5

Passed all tests! ✓

**Question 2**

Correct

Marked out of  
5.00

Given a number N, return true if and only if it is a *confusing number*, which satisfies the following condition:

**Question 3**

Correct

Marked out of  
7.00[Flag question](#)**Objective**

The fundamental data types in C are int, float and char.  
Today, we're discussing int and float data types.

The printf() function prints the given statement to the console. The syntax is printf("format string", argument\_list). In the function, if we are using an integer, character, string or float as argument, then in the format string we have to write %d (integer), %c (character), %s (string), %f (float) respectively.

The scanf() function reads the input data from the console. The syntax is scanf("format string", argument\_list); For ex: The scanf("%d",&number) statement reads integer number from the console and stores the given value in variable **number**.

To input two integers separated by a space on a single line, the command is scanf("%d %d", &n, &m), where **n** and **m** are the two integers.

**Task**

Your task is to take two numbers of [int data type](#), two numbers of float data type as input and output their sum:

1. Declare **4** variables: two of type int and two of type float.
2. Read **2** lines of input from stdin (according to the sequence given in the 'Input Format' section below) and initialize your **4** variables.
3. Use the + and - operator to perform the following operations:
  - o Print the sum and difference of two int variable on a new line.
  - o Print the sum and difference of two float variable rounded to one decimal place on a new line.

**Input Format**

The first line contains two integers.

The second line contains two floating point numbers.

**Constraints**

- **1 ≤ integer variables ≤ 10<sup>4</sup>**
- **1 ≤ float variables ≤ 10<sup>4</sup>**

**Output Format**

Print the sum and difference of both integers separated by a space on the first line, and the sum and difference of both float (scaled to **1** decimal place) separated by a space on the second line.

**Sample Input**

10 4  
4.0 2.0

**Sample Output**

14 6  
6.0 2.0

**Explanation**

When we sum the integers **10** and **4**, we get the integer **14**. When we subtract the second number **4** from the first number **10**, we get **6** as their difference.

Expected	Got
✓ Hello, World!	Hello, World! ✓

Passed all tests! ✓

**Question 2**

Correct

Marked out of  
5.00[Flag question](#)**Objective**

This challenge will help you to learn how to take a character, a string and a sentence as input in C.

To take a single character **ch** as input, you can use `scanf("%c", &ch);` and `printf("%c", ch);` writes a character specified by the argument char to stdout:

```
char ch;
scanf("%c", &ch);
printf("%c", ch);
```

This piece of code prints the character **ch**.

**Task**

You have to print the character, **ch**.

**Input Format**

Take a character, **ch** as input.

**Output Format**

Print the character, **ch**.

**Answer:** (penalty regime: 0 %)

```
1 #include <stdio.h>
2
3 int main () {
4     char ch;
5     scanf("%c", &ch);
6     printf("%c", ch);
7     return 0;
8 }
9 }
```

Input	Expected	Got
✓ C	C	C ✓

Passed all tests! ✓

**Question 3**

Correct

Marked out of  
7.00[Flag question](#)**Objective**

The fundamental data types in C are int, float and char. Today, we're discussing int and float data types.

The `printf()` function prints the given statement to the console. The syntax is `printf("format string", argument_list);`. In the function, if we are using an integer, character, string or float as argument, then in the format string we have to write `%d` (integer), `%c` (character), `%s` (string), `%f` (float) respectively.

The `scanf()` function reads the input data from the console. The syntax is `scanf("format string", argument_list);`. For ex:



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 Tuesday, 8 October 2024, 2:11 PM

Duration 76 days 3 hours

## Question 1

Correct

Marked out of  
3.00

Flag question

## Objective

This is a simple challenge to help you practice printing to stdout.

We're starting out by printing the most famous computing phrase of all time! In the editor below, use either printf or cout to print the string **Hello, World!** to stdout.

## Input Format

You do not need to read any input in this challenge.

## Output Format

Print **Hello, World!** to stdout.

## Sample Output

Hello, World!

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main() {
3     printf("Hello, World!");
4     return 0;
5 }
```

Expected	Got	
✓ Hello, World!	Hello, World!	✓

Passed all tests! ✓

## Question 2

Correct

Marked out of  
5.00

Flag question

## Objective

This challenge will help you to learn how to take a character, a string and a sentence as input in C.

To take a single character **ch** as input, you can



**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
```

Output:

```
WB
BW
BWB
WBW
BWB
```

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 * int main(){
3     int t;
4     scanf("%d", &t);
5     for (int i=0;i<t;i++){
6         int n;
7         char ch;
8         scanf("%d %c", &n, &ch);
9         for(int i=0;i<n;i++){
10            for(int j=0;j<n;j++){
11                if((i+j)%2==0){
12                    printf("%c", ch);
13                }else{
14                    if(ch=='B'){
15                        printf("W");
16                    }else{
17                        printf("B");
18                    }
19            }
20        }
21    }
22 }
```

	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:

## 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** Friday, 29 November 2024, 12:37 PM**Duration** 24 days 4 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;
4     scanf("%d",&t);
5     for(int k=0;k<t;k++){
6         int size;
7         scanf("%d",&size);
8         for(int i=0;i<size;i++){
9             for(int j=0;j<size;j++){
10                 if((i+j)%2==0){
11                     printf("W");
12                 }else{
13                     printf("B");
14                 }
15             }printf("\n");
16     }
17 }
```

	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  
3.00

Let's print a chessboard!

**Explanation 0**

The following sequence of  $n = 2$  food items:

1. Item 1 has 1 macronutrients.
2.  $1 + 2 = 3$ ; observe that this is the max total, and having avoided having exactly  $k = 2$  macronutrients.

**Sample Input 1**

2  
1

**Sample Output 1**

2

**Explanation 1**

1. Cannot use item 1 because  $k = 1$  and  $\text{sum} \equiv k$  has to be avoided at any time.
2. Hence, max total is achieved by  $\text{sum} = 0 + 2 = 2$ .

**Sample Case 2****Sample Input For Custom Testing****Sample Input 2**

3  
3

**Sample Output 2**

5

**Explanation 2**

$2 + 3 = 5$ , is the best case for maximum nutrients.

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main(){
3     long n, k, i, sum = 0;
4     scanf("%ld %ld",&n,&k);
5     for( i = 0 ; i <= n ; i++ ){
6         sum+=i;
7         if( sum == k ){
8             sum -= 1;
9         }
10    }
11    printf("%ld",sum%100000007);
12    return 0;
13 }
```

	Input	Expected	Got	
✓	2 2	3	3	✓
✓	2 1	2	2	✓
✓	3 3	5	5	✓

Passed all tests! ✓

Finish review

**Question 3**

Correct

Marked out of  
7.00[Flag question](#)

A nutritionist is labeling all the best power foods in the market. Every food item arranged in a single line, will have a value beginning from 1 and increasing by 1 for each, until all items have a value associated with them. An item's value is the same as the number of macronutrients it has. For example, food item with value 1 has 1 macronutrient, food item with value 2 has 2 macronutrients, and incrementing in this fashion.

The nutritionist has to recommend the best combination to patients, i.e. maximum total of macronutrients. However, the nutritionist must avoid prescribing a particular sum of macronutrients (an 'unhealthy' number), and this sum is known. The nutritionist chooses food items in the increasing order of their value. Compute the highest total of macronutrients that can be prescribed to a patient, without the sum matching the given 'unhealthy' number.

Here's an illustration:

Given 4 food items (hence value: 1,2,3 and 4), and the unhealthy sum being 6 macronutrients, on choosing items 1, 2, 3 -> the sum is 6, which matches the 'unhealthy' sum. Hence, one of the three needs to be skipped. Thus, the best combination is from among:

- $2 + 3 + 4 = 9$
- $1 + 3 + 4 = 8$
- $1 + 2 + 4 = 7$

Since  $2 + 3 + 4 = 9$ , allows for maximum number of macronutrients, 9 is the right answer.

Complete the code in the editor below. It must return an integer that represents the maximum total of macronutrients, modulo  $1000000007$  ( $10^9 + 7$ ).

It has the following:

*n*: an integer that denotes the number of food items

*k*: an integer that denotes the unhealthy number

**Constraints**

- $1 \leq n \leq 2 \times 10^9$
- $1 \leq k \leq 4 \times 10^{15}$

**Input Format For Custom Testing**

The first line contains an integer, *n*, that denotes the number of food items.

The second line contains an integer, *k*, that denotes the unhealthy number.

**Sample Input 0**

2  
2

**Sample Output 0**

3

**Explanation 0**

The following sequence of  $n = 2$  food items:

1. Item 1 has 1 macronutrients.
2.  $1 + 2 = 3$ ; observe that this is the max total, and having avoided having exactly  $k = 2$  macronutrients.

**Sample Input 1**

2  
1

**Question 2**

Correct

Marked out of  
5.00[Flag question](#)

Given a number N, return true if and only if it is a *confusing number*, which satisfies the following condition:

We can rotate digits by 180 degrees to form new digits. When 0, 1, 6, 8, 9 are rotated 180 degrees, they become 0, 1, 9, 8, 6 respectively. When 2, 3, 4, 5 and 7 are rotated 180 degrees, they become invalid. A *confusing number* is a number that when rotated 180 degrees becomes a **different** number with each digit valid.

**Example 1:**

6 -&gt; 9

Input: 6

Output: true

Explanation:

We get 9 after rotating 6, 9 is a valid number and 9!=6.

**Example 2:**

89 -&gt; 68

Input: 89

Output: true

Explanation:

We get 68 after rotating 89, 86 is a valid number and 86!=89.

**Example 3:**

11 -&gt; 11

Input: 11

Output: false

Explanation:

We get 11 after rotating 11, 11 is a valid number but the value remains the same, thus 11 is not a confusing number.

**Note:**

1.  $0 \leq N \leq 10^9$
2. After the rotation we can ignore leading zeros, for example if after rotation we have 0008 then this number is considered as just 8.

**Answer:** (penalty regime: 0 %)

```

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

	Input	Expected	Got	
✓	6	true	true	✓
✓	89	true	true	✓
✓	25	false	false	✓

Passed all tests! ✓

**Question 3**

Correct

Marked out of  
7.00[Flag question](#)

A nutritionist is labeling all the best power foods in the market. Every food item arranged in a single line, will have a value beginning from 1 and increasing by 1 for each, until all items have a value associated with them. An item's value is the same as the number of macronutrients it has. For example, food item with value 1 has 1 macronutrient, food item with value 2 has 2 macronutrients, and incrementing in

## 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, 12 November 2024, 2:00 PM**Duration** 41 days 3 hours**Question 1**

Correct

Marked out of  
3.00

Flag question

A set of N numbers (separated by one space) is passed as input to the program. The program must identify the count of numbers where the number is odd number.

## Input Format:

The first line will contain the N numbers separated by one space.

## Boundary Conditions:

3 &lt;= N &lt;= 50

The value of the numbers can be from -99999999 to 99999999

## Output Format:

The count of numbers where the numbers are odd numbers.

## Example Input / Output 1:

## Input:

5 10 15 20 25 30 35 40 45 50

## Output:

5

## Explanation:

The numbers meeting the criteria are 5, 15, 25, 35, 45.

**Answer:** (penalty regime: 0 %)

```

1 #include<stdio.h>
2 int main(){
3     int a,count = 0;
4     char ch;
5     while(ch != '\n'){
6         scanf("%d",&a);
7         scanf("%c",&ch);
8         if(a%2!=0){
9             count++;
10        }
11    }
12    printf("%d",count);
13    return 0;
14 }
```

	Input	Expected	Got
✓	5 10 15 20 25 30 35 40 45 50	5	5

Passed all tests! ✓

**Question 2**

Correct

Marked out of  
5.00

Given a number N, return true if and only if it is a *confusing number*, which satisfies the following condition:

**Explanation 0**

The following sequence of  $n = 2$  food items:

1. Item 1 has 1 macronutrients.
2.  $1 + 2 = 3$ ; observe that this is the max total, and having avoided having exactly  $k = 2$  macronutrients.

**Sample Input 1**

2

1

**Sample Output 1**

2

**Explanation 1**

1. Cannot use item 1 because  $k = 1$  and  $\text{sum} \equiv k$  has to be avoided at any time.
2. Hence, max total is achieved by  $\text{sum} = 0 + 2 = 2$ .

**Sample Case 2****Sample Input For Custom Testing****Sample Input 2**

3

3

**Sample Output 2**

5

**Explanation 2**

$2 + 3 = 5$ , is the best case for maximum nutrients.

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main(){
3     long n, k, i, sum = 0;
4     scanf("%ld %ld",&n,&k);
5     for( i = 0 ; i <= n ; i++ ){
6         sum+=i;
7         if( sum == k ){
8             sum -= 1;
9         }
10    }
11    printf("%ld",sum%1000000007);
12    return 0;
13 }
```

	Input	Expected	Got	
✓	2 2	3	3	✓
✓	2 1	2	2	✓
✓	3 3	5	5	✓

Passed all tests! ✓

**Finish review**



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, 12 November 2024, 2:00 PM

Duration 41 days 3 hours

## Question 1

Correct

Marked out of 3.00

[Flag question](#)

A set of N numbers (separated by one space) is passed as input to the program. The program must identify the count of numbers where the number is odd number.

## Input Format:

The first line will contain the N numbers separated by one space.

## Boundary Conditions:

3 &lt;= N &lt;= 50

The value of the numbers can be from -99999999 to 99999999

## Output Format:

The count of numbers where the numbers are odd numbers.

## Example Input / Output 1:

## Input:

5 10 15 20 25 30 35 40 45 50

## Output:

5

## Explanation:

The numbers meeting the criteria are 5, 15, 25, 35, 45.

## Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main(){
3     int a,count = 0;
4     char ch;
5     while(ch != '\n'){
6         scanf("%d",&a);
7         scanf("%c",&ch);
8         if(a%2!=0){
9             count++;
10        }
11    }
12    printf("%d",count);
13 }
14 }
```



Refer the sample output for formatting

Sample Input 1:

10

Sample Output 1:

4

Sample Input 2:

5

Sample Output 2:

3

Explanation:

For test case 1, N=10.

According to Manish {1, 2, 3, ..., 10} must be distributed.

But as per Manisha only {1, 2, 3, 4} coins are enough to purchase any item ranging from \$1 to \$10. Hence minimum is 4. Likewise denominations could also be {1, 2, 3, 5}. Hence answer is still 4.

For test case 2, N=5.

According to Manish {1, 2, 3, 4, 5} must be distributed.

But as per Manisha only {1, 2, 3} coins are enough to purchase any item ranging from \$1 to \$5. Hence minimum is 3. Likewise, denominations could also be {1, 2, 4}. Hence answer is still 3.

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main(){
3     int maxprice;
4     scanf("%d",&maxprice);
5     int count = 0;
6     int demo = 1;
7     while(demo<=maxprice){
8         maxprice-=demo;
9         demo*=2;
10        count++;
11    }
12    if(maxprice>0){
13        count++;
14    }
15    printf("%d",count);
16    return 0;
17 }
```

	Input	Expected	Got	
✓	10	4	4	✓
✓	5	3	3	✓
✓	20	5	5	✓
✓	500	9	9	✓
✓	1000	10	10	✓

Passed all tests! ✓

Finish review

**Question 3**

Correct

Marked out of  
7.00[Flag question](#)

The problem solvers have found a new Island for coding and named it as Philaland. These smart people were given a task to make a purchase of items at the Island easier by distributing various coins with different values. Manish has come up with a solution that if we make coins category starting from \$1 till the maximum price of the item present on Island, then we can purchase any item easily. He added the following example to prove his point.

Let's suppose the maximum price of an item is 5\$ then we can make coins of { \$1, \$2, \$3, \$4, \$5} to purchase any item ranging from \$1 till \$5.

Now Manisha, being a keen observer suggested that we could actually minimize the number of coins required and gave following distribution { \$1, \$2, \$3}. According to him any item can be purchased one time ranging from \$1 to \$5. Everyone was impressed with both of them. Your task is to help Manisha come up with a minimum number of denominations for any arbitrary max price in Philaland.

**Input Format**

Contains an integer N denoting the maximum price of the item present on Philaland.

**Output Format**

Print a single line denoting the minimum number of denominations of coins required.

**Constraints**

1 &lt;= T &lt;= 100

1 &lt;= N &lt;= 5000

**Refer the sample output for formatting****Sample Input 1:**

10

**Sample Output 1:**

4

**Sample Input 2:**

5

**Sample Output 2:**

3

**Explanation:**

For test case 1, N=10.

According to Manish { \$1, \$2, \$3... \$10} must be distributed.

But as per Manisha only { \$1, \$2, \$3, \$4} coins are enough to purchase any item ranging from \$1 to \$10. Hence minimum is 4. Likewise denominations could also be { \$1, \$2, \$3, \$5}. Hence answer is still 4.

For test case 2, N=5.

According to Manish { \$1, \$2, \$3, \$4, \$5} must be distributed.

But as per Manisha only { \$1, \$2, \$3} coins are enough to purchase any item ranging from \$1 to \$5. Hence minimum is 3. Likewise, denominations could also be { \$1, \$2, \$4}. Hence

1 2 3 3

**Sample Output 0**

NO

YES

**Explanation 0**

For the first test case, no such index exists.

For the second test case,  $\text{arr}[0] + \text{arr}[1] = \text{arr}[3]$ , therefore index 2 satisfies the given conditions.

**Sample Input 1**

```
3
5
1 1 4 1 1
4
2 0 0 0
4
0 0 2 0
```

**Sample Output 1**

```
YES
YES
YES
```

**Explanation 1**

In the first test case,  $\text{arr}[2] = 4$  is between two subarrays summing to 2.

In the second case,  $\text{arr}[0] = 2$  is between two subarrays summing to 0.

In the third case,  $\text{arr}[2] = 4$  is between two subarrays summing to 0.

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main(){
3     int t, n, is, rs, m;
4     scanf("%d",&t);
5     for(int i = 0;i<t;i++){
6         is=0;
7         rs=0;
8         scanf("%d",&n);
9         int arr[n];
10        for(int j=0;j<n;j++)
11            scanf("%d",&arr[j]);
12        m = n/2;
13        if(arr[m] == 0){
14            for(m = 0;arr[m]==0 && m<n;m++)
15            }
16            for (int j = 0;j<=m;j++)
17                is = is + arr[j];
18            for(int j = m;j<n;j++)
19                rs = rs+arr[j];
20            printf("%s\n",is == rs) ? "YES"
21        }
22    }
```

	Input	Expected	Got	
✓	3 5 1 1 4 1 1 4 2 0 0 0 4 0 0 2 0	YES YES YES YES	YES YES YES	✓
✓	2 3 1 2 3 4 1 2 3 3	NO YES	NO YES	✓
Passed all tests! ✓				

of holes for all of its digits. For example, the number 819 has 3 holes.

Complete the program, it must return an integer denoting the total number of holes in num.

#### Constraints

1 ≤ num ≤ 10<sup>9</sup>

#### Input Format For Custom Testing

There is one line of text containing a single integer num, the value to process.

#### Sample Input

630

#### Sample Output

2

#### Explanation

Add the holes count for each digit, 6, 3 and 0. Return 1 + 0 + 1 = 2.

#### Sample Case 1

#### Sample Input

1288

#### Sample Output

4

#### Explanation

Add the holes count for each digit, 1, 2, 8, 8. Return 0 + 0 + 2 + 2 = 4.

#### Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int countholes(int digit){
3     switch(digit){
4         case 0:
5         case 4:
6         case 6:
7         case 9:
8             return 1;
9         case 8:
10            return 2;
11        default:
12            return 0;
13    }
14 }
15 int main (){
16     long long number;
17     scanf("%lld",&number);
18     int totalholes=0;
19     while(number>0){
20         int digit = number%10;
21         totalholes += countholes(digit);
22         number/=10;
23     }printf("%d",totalholes);
24     return 0;
25 }
```

	Input	Expected	Got	
✓	630	2	2	✓
✓	1288	4	4	✓

Passed all tests! ✓

**Question 2**

Correct

Marked out of  
5.00[Flag question](#)

You are designing a poster which prints out numbers with a unique style applied to each of them. The styling is based on the number of closed paths or holes present in a given number.

The number of holes that each of the digits from 0 to 9 have are equal to the number of closed paths in the digit. Their values are:

1, 2, 3, 5, and 7 = 0 holes.

0, 4, 6, and 9 = 1 hole.

8 = 2 holes.

Given a number, you must determine the sum of the number of holes for all of its digits. For example, the number 819 has 3 holes.

Complete the program, it must return an integer denoting the total number of holes in num.

**Constraints**

$1 \leq \text{num} \leq 10^9$

**Input Format For Custom Testing**

There is one line of text containing a single integer num, the value to process.

**Sample Input**

630

**Sample Output**

2

**Explanation**

Add the holes count for each digit, 6, 3 and 0. Return  $1 + 0 + 1 = 2$ .

**Sample Case 1****Sample Input**

1288

**Sample Output**

4

**Explanation**

Add the holes count for each digit, 1, 2, 8, 8. Return  $0 + 0 + 2 + 2 = 4$ .

**Answer: (penalty regime: 0 %)**

```
1 #include<stdio.h>
2 *
3 * int countholes(int digit){
4 *     switch(digit){
5 *         case 0:
6 *         case 4:
7 *         case 6:
8 *         case 9:
9 *             return 1;
10 *        case 8:
11 *            return 2;
12 *        default:
13 *            return 0;
14 *     }
15 * }
16 * int main (){
17 *     long long number;
18 *     scanf("%lld",&number);
19 *     int totalholes=0;
20 *     while(number>0){
```

REC-CIS

## Quiz navigation

1    2    3

Show one page at a time

Finish review

Status Finished

Started Monday, 23 December 2024, 5:33 PM

Completed Friday, 8 November 2024, 1:12 PM

Duration 45 days 4 hours

## Question 1

Correct

Marked out of  
3.00

Flag question

Alice and Bob are playing a game called "Stone Game". Stone game is a two-player game. Let N be the total number of stones. In each turn, a player can remove either one stone or four stones. The player who picks the last stone, wins. They follow the "Ladies First" norm. Hence Alice is always the one to make the first move. Your task is to find out whether Alice can win, if both play the game optimally.

## Input Format

First line starts with T, which is the number of test cases.  
Each test case will contain N number of stones.

## Output Format

Print "Yes" in the case Alice wins, else print "No".

## Constraints

1&lt;=T&lt;=1000

1&lt;=N&lt;=10000

## Sample Input and Output

## Input

```
3
1
6
7
```

## Output

```
Yes
Yes
No
```

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main () {
3     int T,N;
4     scanf ("%d",&T);
5     while(T!=0){
6         scanf ("%d",&N);
7         if((N%4 + N/4)%2==0){
8             printf("No\n");
9         }else{
10            printf("Yes\n");
11        }
12        T=T-1;
13    }
14 }
```

	Input	Expected	Got	
✓	3	Yes	Yes	✓
	1	Yes	Yes	
	6	No	No	
	7			

Passed all tests! ✓



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 Friday, 8 November 2024, 1:12 PM

Duration 45 days 4 hours

## Question 1

Correct

Marked out of  
3.00[Flag question](#)

Alice and Bob are playing a game called "Stone Game". Stone game is a two-player game. Let N be the total number of stones. In each turn, a player can remove either one stone or four stones. The player who picks the last stone, wins. They follow the "Ladies First" norm. Hence Alice is always the one to make the first move. Your task is to find out whether Alice can win, if both play the game optimally.

## Input Format

First line starts with T, which is the number of test cases.  
Each test case will contain N number of stones.

## Output Format

Print "Yes" in the case Alice wins, else print "No".

## Constraints

1&lt;=T&lt;=1000

1&lt;=N&lt;=10000

## Sample Input and Output

## Input

3  
1  
6  
7

## Output

Yes  
Yes  
No

## Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main () {
3     int T,N;
4     scanf("%d",&T);
5     while(T!=0){
6         scanf("%d",&N);
7         if((N%4 + N/4)%2==0){
8             printf("No\n");
9         }else{
10            printf("Yes\n");
11        }
12    }
13 }
14 }
```





	Input	Expected	Got	
✓	T 10 20	200	200	✓
✓	S 30 40	600	600	✓
✓	B 2 11	0	0	✓
✓	R 10 30	300	300	✓
✓	S 40 50	1000	1000	✓

Passed all tests! ✓

**Question 3**

Correct

Marked out of  
7.00[Flag question](#)

Superman is planning a journey to his home planet. It is very important for him to know which day he arrives there. They don't follow the 7-day week like us. Instead, they follow a 10-day week with the following days: Day Number Name of Day  
 1 Sunday 2 Monday 3 Tuesday 4 Wednesday 5 Thursday 6 Friday 7 Saturday 8 Kryptoday 9 Coluday 10 Daxamday  
 Here are the rules of the calendar: • The calendar starts with Sunday always. • It has only 296 days. After the 296th day, it goes back to Sunday. You begin your journey on a Sunday and will reach after n. You have to tell on which day you will arrive when you reach there.

Input format:

Contain a number n (0 &lt; n)

Output format: Print the name of the day you are arriving on

Example Input

7

Example Output

Kryptoday

Example Input

1

Example Output Monday

**Answer:** (penalty regime: 0 %)

```

1 #include<stdio.h>
2 int main(){
3     int n;
4     scanf("%d",&n);
5     char*day[]={"Sunday","Monday","Tuesday"
6     int a=(n%296)%10;
7     printf("%s",day[a]);
8     return 0;
9 }
```

	Input	Expected	Got	
✓	7	Kryptoday	Kryptoday	✓
✓	1	Monday	Monday	✓

Passed all tests! ✓

[Finish review](#)



REC-CIS

G  
8  
8

Sample Output 4

0

Sample Input

C  
9  
10

Sample Output 4

0

Explanation:

- First is output of area of rectangle
- Then, output of area of triangle
- Then output of area square
- Finally, something random, so we print 0

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main(){
3     int a,b;
4     char c;
5     scanf("%c\n%d\n%d", &c, &a, &b);
6     if(c=='T'){
7         printf("%d", a*b);
8     }else if(c=='R'){
9         printf("%d", a*b);
10    }else if(c=='S'){
11        printf("%d", (a*b)/2);
12    }else{
13        printf("%d", 0);
14    }
15 }
```

	Input	Expected	Got	
✓	T 10 20	200	200	✓
✓	S 30 40	600	600	✓
✓	B 2 11	0	0	✓
✓	R 10 30	300	300	✓
✓	S 40 50	1000	1000	✓

Passed all tests! ✓



**Question 2**

Correct

Marked out of  
5.00[Flag question](#)

Suppandi is trying to take part in the local village math quiz. In the first round, he is asked about shapes and areas. Suppandi, is confused, he was never any good at math. And also, he is bad at remembering the names of shapes. Instead, you will be helping him calculate the area of shapes.

- When he says rectangle he is actually referring to a square.
- When he says square, he is actually referring to a triangle.
- When he says triangle he is referring to a rectangle
- And when he is confused, he just says something random. At this point, all you can do is say 0.

Help Suppandi by printing the correct answer in an integer.

**Input Format**

- Name of shape (always in upper case R à Rectangle, S à Square, T à Triangle)
- Length of 1 side
- Length of other side

Note: In case of triangle, you can consider the sides as height and length of base

**Output Format**

- Print the area of the shape.

**Sample Input 1**

T

10

20

**Sample Output 1**

200

**Sample Input 2**

S

30

40

**Sample Output 2**

600

**Sample Input 3**

R

10

10

**Sample Output 3**

100

**Sample Input 4**

G

8

8

**Sample Output 4**

0



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	Monday, 28 October 2024, 2:49 PM
Duration	56 days 2 hours

Question 1  
Correct  
Marked out of 3.00  
[Flag question](#)

Some data sets specify dates using the year and day of year rather than the year, month, and day of month. The day of year (DOY) is the sequential day number starting with day 1 on January 1st.

There are two calendars - one for normal years with 365 days, and one for leap years with 366 days. Leap years are divisible by 4. Centuries, like 1900, are not leap years unless they are divisible by 400. So, 2000 was a leap year.

To find the day of year number for a standard date, scan down the Jan column to find the day of month, then scan across to the appropriate month column and read the day of year number. Reverse the process to find the standard date for a given day of year.

Write a program to print the Day of Year of a given date, month and year.

## Sample Input 1

18  
6  
2020

## Sample Output 1

170

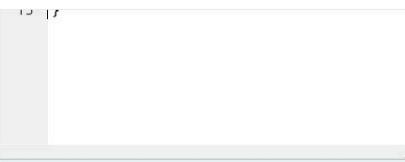
Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main() {
3     int a,b,c,count=0;
4     scanf("%d\n%d\n%d",&a,&b,&c);
5     for (int i=1; i<b ; i++){
6         if(i%2==0){
7             count=count+31;
8         }else if(i==2){
9             if(c%4==0 && c%100!=0){
10                 count=count+29;
11             }else if(c%400 == 0 && c%100!=0){
12                 count=count+29;
13             }else{
14                 count=count+28;
15             }
16         }else if(i%2==0){
17             count=count+30;
18         }
19     }printf("%d",count+a);
20 }
21 }
```

	Input	Expected	Got	
✓	18 6 2020	170	170	✓

Passed all tests! ✓





	Input	Expected	Got	
✓	T 10 20	200	200	✓
✓	S 30 40	600	600	✓
✓	B 2 11	0	0	✓
✓	R 10 30	300	300	✓
✓	S 40 50	1000	1000	✓

Passed all tests! ✓

**Question 3**

Correct

Marked out of  
7.00

Flag question

Superman is planning a journey to his home planet. It is very important for him to know which day he arrives there. They don't follow the 7-day week like us. Instead, they follow a 10-day week with the following days: Day Number Name of Day  
 1 Sunday 2 Monday 3 Tuesday 4 Wednesday 5 Thursday 6 Friday 7 Saturday 8 Kryptonday 9 Coluday 10 Daxamday  
 Here are the rules of the calendar:

- The calendar starts with Sunday always.
- It has only 296 days. After the 296th day, it goes back to Sunday.
- You begin your journey on a Sunday and will reach after n. You have to tell on which day you will arrive when you reach there.

Input format:

Contain a number n (0 &lt; n)

Output format: Print the name of the day you are arriving on

Example Input

7

Example Output

Kryptonday

Example Input

1

Example Output Monday

**Answer:** (penalty regime: 0 %)

```

1 #include<stdio.h>
2 int main(){
3     int n;
4     scanf("%d",&n);
5     char*day[]={"Sunday", "Monday", "Tuesday",
6     int a=(n%296)%10;
7     printf("%s",day[a]);
8     return 0;
9 }
```

	Input	Expected	Got	
✓	7	Kryptonday	Kryptonday	✓
✓	1	Monday	Monday	✓

Passed all tests! ✓

Finish review

10

3

**Sample Output 0**

5

**Explanation 0**

Factoring  $n = 10$  we get  $\{1, 2, 5, 10\}$ . We then return the  $p = 3^{\text{rd}}$  factor as our answer.

**Sample Input 1**

10

5

**Sample Output 1**

0

**Explanation 1**

Factoring  $n = 10$  we get  $\{1, 2, 5, 10\}$ . There are only 4 factors and  $p = 5$ . We return 0 as our answer.

**Sample Input 2**

1

1

**Sample Output 2**

1

**Explanation 2**

Factoring  $n = 1$  we get  $\{1\}$ . We then return the  $p = 1^{\text{st}}$  factor as our answer.

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main(){
3     int a, b;
4     scanf("%d", &a);
5     scanf("%d", &b);
6     int c[a], d = 0;
7     for(int i = 1; i <= a; i++){
8         if((a % i) == 0){
9             c[d] = i;
10            d++;
11            if(d == b){
12                break;
13            }
14        }
15    }
16    (d < b) ? printf("0") : printf("%d",
17 }
```

	Input	Expected	Got	
✓	10 3	5	5	✓
✓	10 5	0	0	✓
✓	1 1	1	1	✓

Passed all tests! ✓

**Finish review**

10

10

Sample Output 3

100

Sample Input 4

G

8

8

Sample Output 4

0

Sample Input

C

9

10

Sample Output 4

0

Explanation:

- First is output of area of rectangle
- Then, output of area of triangle
- Then output of area square
- Finally, something random, so we print 0

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main(){
3     int a,b;
4     char c;
5     scanf("%c\n%d\n%d", &c, &a, &b);
6     if(c=='T'){
7         printf("%d", a*b);
8     }else if(c=='R'){
9         printf("%d", a*b);
10    }else if(c=='S'){
11        printf("%d", (a*b)/2);
12    }else{
13        printf("%d", 0);
14    }
15 }
```

	Input	Expected	Got	
✓	T 10 20	200	200	✓
✓	S 30 40	600	600	✓
✓	B 2 11	0	0	✓
✓	R 10 30	300	300	✓
✓	S 40 50	1000	1000	✓

Passed all tests! ✓

**Question 3**

Correct

Marked out of  
1.00[Flag question](#)

Determine all positive integer values that evenly divide into a number, its factors. Return the  $p^{\text{th}}$  element of your list, sorted ascending. If there is no  $p^{\text{th}}$  element, return 0.

For example, given the number  $n = 20$ , its factors are  $\{1, 2, 4, 5, 10, 20\}$ . Using **1-based indexing** if  $p = 3$ , return 4. If  $p > 6$ , return 0.

Complete the code in the editor below. The function should return a long integer value of the  $p^{\text{th}}$  integer factor of  $n$ .

It has the following:

$n$ : an integer

$p$ : an integer

**Constraints**

- $1 \leq n \leq 10^{15}$
- $1 \leq p \leq 10^9$

**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 to factor.

The second line contains an integer  $p$ , the 1-based index of the factor to return.

**Sample Input 0**

```
10
3
```

**Sample Output 0**

```
5
```

**Explanation 0**

Factoring  $n = 10$  we get  $\{1, 2, 5, 10\}$ . We then return the  $p = 3^{\text{rd}}$  factor as our answer.

**Sample Input 1**

```
10
5
```

**Sample Output 1**

```
0
```

**Explanation 1**

Factoring  $n = 10$  we get  $\{1, 2, 5, 10\}$ . There are only 4 factors and  $p = 5$ . We return 0 as our answer.

**Sample Input 2**

```
1
1
```

**Sample Output 2**

```
1
```

**Explanation 2**

Factoring  $n = 1$  we get  $\{1\}$ . We then return the  $p = 1^{\text{st}}$  factor as our answer.

**Question 2**

Correct

Marked out of  
5.00[Flag question](#)

Suppandi is trying to take part in the local village math quiz. In the first round, he is asked about shapes and areas. Suppandi, is confused, he was never any good at math. And also, he is bad at remembering the names of shapes. Instead, you will be helping him calculate the area of shapes.

- When he says rectangle he is actually referring to a square.
- When he says square, he is actually referring to a triangle.
- When he says triangle he is referring to a rectangle
- And when he is confused, he just says something random. At this point, all you can do is say 0.

Help Suppandi by printing the correct answer in an integer.

**Input Format**

- Name of shape (always in upper case R à Rectangle, S à Square, T à Triangle)
- Length of 1 side
- Length of other side

Note: In case of triangle, you can consider the sides as height and length of base

**Output Format**

- Print the area of the shape.

**Sample Input 1**

T  
10  
20

**Sample Output 1**

200

**Sample Input 2**

S  
30  
40

**Sample Output 2**

600

**Sample Input 3**

R  
10  
10

**Sample Output 3**

100

**Sample Input 4**

G  
8  
8

**Sample Output 4**



## Week-03-03-Practice Ses...



rajalakshmicolleges.org

REC-CIS

Finish review

## Question 1

Correct

Marked out of  
3.00[Flag question](#)

Some data sets specify dates using the year and day of year rather than the year, month, and day of month. The day of year (DOY) is the sequential day number starting with day 1 on January 1st.

There are two calendars - one for normal years with 365 days, and one for leap years with 366 days. Leap years are divisible by 4. Centuries, like 1900, are not leap years unless they are divisible by 400. So, 2000 was a leap year.

To find the day of year number for a standard date, scan down the Jan column to find the day of month, then scan across to the appropriate month column and read the day of year number. Reverse the process to find the standard date for a given day of year.

Write a program to print the Day of Year of a given date, month and year.

## Sample Input 1

```
18
6
2020
```

## Sample Output 1

```
170
```

## Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main() {
3     int a,b,c,count=0;
4     scanf("%d\n%d\n%d", &a,&b,&c);
5     for (int i= 1; i<b ; i++){
6         if(i%2!=0){
7             count=count+31;
8         }else if(i==2){
9             if(c%4==0 && c%100!=0){
10                 count=count+29;
11             }else if(c%400 == 0 && c%100!=0){
12                 count=count+29;
13             }else{
14                 count=count+28;
15             }
16         }else if(i%2==0){
17             count=count+30;
18         }
19     }printf("%d",count+a);
20 }
21 }
```

	Input	Expected	Got	
✓	18 6 2020	170	170	✓

Passed all tests! ✓

## Question 2

Correct

Marked out of  
5.00[Flag question](#)

Suppandi is trying to take part in the local village math quiz. In the first round, he is asked about shapes and areas. Suppandi, is confused, he was never any good at math. And also, he is bad at remembering the names of shapes. Instead, you will be helping him calculate the area of shapes.

When he says rectangle he is actually referring to a



✓	2004	monkey	monkey	✓
✓	2010	Tiger	Tiger	✓

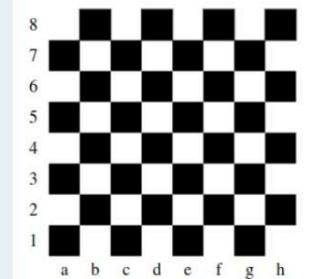
Passed all tests! ✓

**Question 3**

Correct

Marked out of  
7.00[Flag question](#)

Positions on a chess board are identified by a letter and a number. The letter identifies the column, while the number identifies the row, as shown below:



Write a program that reads a position from the user. Use an if statement to determine if the column begins with a black square or a white square. Then use modular arithmetic to report the color of the square in that row. For example, if the user enters a1 then your program should report that the square is black. If the user enters d5 then your program should report that the square is white. Your program may assume that a valid position will always be entered. It does not need to perform any error checking.

**Sample Input 1**

a 1

**Sample Output 1**

The square is black.

**Sample Input 2**

d 5

**Sample Output 2**

The square is white.

**Answer:** (penalty regime: 0 %)

```

1 #include<stdio.h>
2 int main (){
3     int a;
4     char b;
5     scanf("%c %d",&b,&a);
6     if(a%2==0){
7         if(b=='b' || b=='d' || b=='f' || b=='h')
8             printf("The square is black.");
9         else{
10            printf("The square is white.");
11        }
12     }else{
13         if(b=='c' || b=='e' || b=='g')
14             printf("The square is white.");
15         else{
16            printf("The square is black.");
17        }
18    }
19 }
```

	Input	Expected	Got
✓	a 1	The square is black.	The square is bl
✓	d 5	The square is white.	The square is wh

Passed all tests! ✓

[Finish review](#)

**Question 3**

Correct

Marked out of  
5.00[Flag question](#)

Watson gives Sherlock an array of integers. His challenge is to find an element of the array such that the sum of all elements to the left is equal to the sum of all elements to the right. For instance, given the array  $arr = [5, 6, 8, 11]$ , 8 is between two subarrays that sum to 11. If your starting array is [1], that element satisfies the rule as left and right sum to 0.

You will be given arrays of integers and must determine whether there is an element that meets the criterion.

Complete the code in the editor below. It should return a string, either YES if there is an element meeting the criterion or NO otherwise.

It has the following:

- arr: an array of integers

**Input Format**

The first line contains  $T$ , the number of test cases.

The next  $T$  pairs of lines each represent a test case.

- The first line contains  $n$ , the number of elements in the array  $arr$ .
- The second line contains  $n$  space-separated integers  $arr[i]$  where  $0 \leq i < n$ .

**Constraints**

- $1 \leq T \leq 10$
- $1 \leq n \leq 10^5$
- $1 \leq arr[i] \leq 2 \times 10^4$
- $0 \leq i \leq n$

**Output Format**

For each test case print YES if there exists an element in the array, such that the sum of the elements on its left is equal to the sum of the elements on its right; otherwise print NO.

**Sample Input 0**

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

**Sample Output 0**

```
NO
YES
```

**Explanation 0**

For the first test case, no such index exists.

For the second test case,  $arr[0] + arr[1] = arr[3]$ , therefore index 2 satisfies the given conditions.

**Sample Input 1**

```
3
5
1 1 4 1 1
4
2 0 0 0
4
0 0 2 0
```

**Sample Output 1**

**Input Format**

There will be four lines of input:

*n* - the size of the first list, *arr*

The next line contains *n* space-separated integers *arr[i]*

*m* - the size of the second list, *brr*

The next line contains *m* space-separated integers *brr[i]*

**Constraints**

- $1 \leq n, m \leq 2 \times 10^5$
- $n \leq m$
- $1 \leq brr[i] \leq 2 \times 10^4$
- $X_{\max} - X_{\min} < 101$

**Output Format**

Output the missing numbers in ascending order.

**Sample Input**

```
10
203 204 205 206 207 208 203 204 205 206
13
203 204 204 205 206 207 205 208 203 206 205 206 204
```

**Sample Output**

```
204 205 206
```

**Explanation**

**204** is present in both arrays. Its frequency in *arr* is **2**, while its frequency in *brr* is **3**. Similarly, **205** and **206** occur twice in *arr*, but three times in *brr*. The rest of the numbers have the same frequencies in both lists.

**Answer: (penalty regime: 0 %)**

```
1 #include<stdio.h>
2 int main(){
3     int a;
4     scanf("%d",&a);
5     int arr1[a];
6     for(int i = 0;i < a;i++){
7         scanf("%d",&arr1[i]);
8     }int b;
9     scanf("%d",&b);
10    int arr2[b];
11    for(int i = 0;i < b;i++){
12        scanf("%d",&arr2[i]);
13    }for(int i = 0;i < a;i++){
14        for(int j = 0;j < b;j++){
15            if(arr1[i] == arr2[j]){
16                arr1[i] = -1;
17                arr2[j] = -1;
18            }
19        }
20    }for(int i = 0;i < b;i++){
21        for(int j = 0; j < b-1-i;j++){
22            if(arr2[j] > arr2[j+1]){
23                int temp = arr2[j+1];
24                arr2[j+1] = arr2[j];
25                arr2[j] = temp;
26            }
27        }
28    }for(int i = 0;i < b;i++){
29        if(arr2[i] != -1){
30            printf("%d ",arr2[i]);
31        }
32    }
33 }
```

	Input
✓	10 203 204 205 206 207 208 203 204 205 206 13 203 204 204 205 206 207 205 208 203 206 205

Passed all tests! ✓

**Question 2**

Correct

Marked out of  
5.00[Flag question](#)

Numeros the Artist had two lists that were permutations of one another. He was very proud. Unfortunately, while transporting them from one exhibition to another, some numbers were lost out of the first list. Can you find the missing numbers?

As an example, the array with some numbers missing, **arr** = [7, 2, 5, 3, 5, 3]. The original array of numbers **brr** = [7, 2, 5, 4, 6, 3, 5, 3]. The numbers missing are [4, 6].

**Notes**

- If a number occurs multiple times in the lists, you must ensure that the frequency of that number in both lists is the same. If that is not the case, then it is also a missing number.
- You have to print all the missing numbers in ascending order.
- Print each missing number once, even if it is missing multiple times.
- The difference between maximum and minimum number in the second list is less than or equal to **100**.

Complete the code in the editor below. It should return an array of missing numbers.

It has the following:

- arr**: the array with missing numbers
- brr**: the original array of numbers

**Input Format**

There will be four lines of input:

**n** - the size of the first list, **arr**

The next line contains **n** space-separated integers **arr[i]**

**m** - the size of the second list, **brr**

The next line contains **m** space-separated integers **brr[j]**

**Constraints**

- $1 \leq n, m \leq 2 \times 10^5$
- $n \leq m$
- $1 \leq brr[j] \leq 2 \times 10^4$
- $X_{\max} - X_{\min} < 101$

**Output Format**

Output the missing numbers in ascending order.

**Sample Input**

```
10
203 204 205 206 207 208 203 204 205 206
13
203 204 204 205 206 207 205 208 203 206 205 206 204
```

**Sample Output**

```
204 205 206
```

**Explanation**

**204** is present in both arrays. Its frequency in **arr** is **2**, while its frequency in **brr** is **3**. Similarly, **205** and **206** occur twice in **arr**, but three times in **brr**. The rest of the numbers have the same frequencies in both lists.

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main(){
3     int a;
```

**Constraints**

- $1 \leq t \leq 50$
- $2 \leq m \leq 10^4$
- $2 \leq n \leq 10^4$
- $1 \leq \text{cost}[i] \leq 10^4$ , " $i \in [1, n]$ "
- There will always be a unique solution.

**Output Format**

For each test case, print two space-separated integers denoting the indices of the two flavors purchased, in ascending order.

**Sample Input**

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

**Sample Output**

```
1 4
1 2
```

**Explanation**

Sunny and Johnny make the following two trips to the parlor:

- The first time, they pool together  $m = 4$  dollars. Of the five flavors available that day, flavors **1** and **4** have a total cost of  $1 + 3 = 4$ .
- The second time, they pool together  $m = 4$  dollars. Of the four flavors available that day, flavors **1** and **2** have a total cost of  $2 + 2 = 4$ .

**Answer:** (penalty regime: 0 %)

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

	Input	Expected	Got	
✓	2 4 5 1 4 5 3 2 4 4 2 2 4 3	1 4 1 2      	1 4 1 2       	✓

Passed all tests! ✓



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 Friday, 20 December 2024, 1:07 PM

Duration 3 days 4 hours

## Question 1

Correct

Marked out of 5.00

[Flag question](#)

Sunny and Johnny like to pool their money and go to the ice cream parlor. Johnny never buys the same flavor that Sunny does. The only other rule they have is that they spend all of their money.

Given a list of prices for the flavors of ice cream, select the two that will cost all of the money they have.

For example, they have  $m = 6$  to spend and there are flavors costing  $\text{cost} = [1, 2, 3, 4, 5, 6]$ . The two flavors costing 1 and 5 meet the criteria. Using 1-based indexing, they are at indices 1 and 4.

## Function Description

Complete the code in the editor below. It should return an array containing the indices of the prices of the two flavors they buy.

It has the following:

- $m$ : an integer denoting the amount of money they have to spend
- $\text{cost}$ : an integer array denoting the cost of each flavor of ice cream

## Input Format

The first line contains an integer,  $t$ , denoting the number of trips to the ice cream parlor. The next  $t$  sets of lines each describe a visit. Each trip is described as follows:

1. The integer  $m$ , the amount of money they have pooled.
2. The integer  $n$ , the number of flavors offered at the time.
3.  $n$  space-separated integers denoting the cost of each flavor:  $\text{cost}[\text{cost}[1], \text{cost}[2], \dots, \text{cost}[n]]$ .

**Note:** The index within the cost array represents the flavor of the ice cream purchased.

## Constraints

- $1 \leq t \leq 50$
- $2 \leq m \leq 10^4$
- $2 \leq n \leq 10^4$
- $1 \leq \text{cost}[i] \leq 10^4$ , " $i \in [1, n]$ "
- There will always be a unique solution.

## Output Format

For each test case, print two space-separated integers denoting the indices of the two flavors purchased, in ascending order.

## Sample Input

