CHALLENGE INFORMATION

You have already solved this challenge ! Though you can run the code with different logic !

Course	JAVA	Session	Datatypes and Operators	Question Information	Level 2 Challenge 14

Problem

Question description

This winter is so cold in Sweden! A group of n friends decided to buy k bottles of a soft drink called "Take-It-Light" to warm up a bit. Each bottle has l milliliters of the drink. Also they bought c limes and cut each of them into d slices. After that they found p grams of salt.

To make a toast, each friend needs nl milliliters of the drink, a slice of lime and np grams of salt. The friends want to make as many toasts as they can, provided they all drink the same amount. How many toasts can each friend make?

Constraints:

$1 \le n, k, l, c, d, p, nl, np \le 1000$

Input Format:

The first and only line contains positive integers n, k, l, c, d, p, nl, np, not exceeding 1000 and no less than 1. The numbers are separated by exactly one space.

Output Format:

Print a single integer — the number of toasts each friend can make.

Test Cases

Logical Test Cases

Test Case 1

INPUT (STDIN)

```
3 4 5 10 8 100 3 1
```

EXPECTED OUTPUT

```
2
```

Test Case 2

INPUT (STDIN)

```
5 100 10 1 19 90 4 3
```

EXPECTED OUTPUT

```
3
```

Mandatory Test Cases

Test Case 1

KEYWORD

```
int n,k,l,c,d,p,nl,np;
```

Test Case 2

KEYWORD

```
l=sc.nextInt();
```

Test Case 3

KEYWORD

```
Math.min
```

Complexity Test Cases

Test Case 1

CYCLOMATIC COMPLEXITY

```
1
```

Test Case 2

TOKEN COUNT

```
172
```

Test Case 3

NLOC

```
23
```

Code Editor

You have already solved this challenge ! Though you can run the code with different logic !

Code Editor | JAVA SE 1.8 | Light Theme

```java
import java.util.Scanner;

public class Class332241010280 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n,k,l,c,d,p,nl,np;
         n = sc.nextInt();
         k = sc.nextInt();
         l=sc.nextInt();
         c = sc.nextInt();
        d = sc.nextInt();
        p = sc.nextInt(); nl = sc.nextInt();
        np = sc.nextInt();
        int minToasts = Math.min((k * l) / nl, Math.min(c *
        System.out.println(minToasts / n);
    }
}
```

SAVE | RESET | RUN | EVALUATE

Custom Input (stdin) | T1 | T2

Type Here

Output | MATCH T1 | MATCH T2

Empty

Complexity Analysis

Test Case Status