

# Mid Exam Preparation – 12 June 2023

## 1. Bonus Scoring System

Link: <https://judge.softuni.bg/Contests/Practice/Index/2028#0>

Create a program that calculates **bonus points** for each **student**, for a certain **course**. On the first line, you are going to receive **the count of the students** for this course. **On the second line**, you will receive **the count of the lectures** in the course. Every course has an **additional bonus**. You are going to receive it **on the third line**. On the next lines, you will be receiving the **count of attendances for each student**.

The bonus is calculated with the following **formula**:

$\{\text{total bonus}\} = \{\text{student attendances}\} / \{\text{course lectures}\} * (5 + \{\text{additional bonus}\})$

Find the student with the **maximum bonus** and print him/her, along with **his attendances** in the following format:

"Max Bonus: {maxBonusPoints}."

"The student has attended {studentAttendances} lectures."

Round the bonus points at the end to **the nearest bigger number**.

## Input / Constrains

- On the **first line** you are going to receive the count of the students – an integer number in the range [0...50]
- On the **second line** you are going to receive the count of the lectures – an integer number in the range [0...50].
- On the **third line** you are going to receive **the initial bonus** – an integer number in the range [0....100].
- On the next lines**, you will be receiving the **attendances of each student**.
- There will **never** be **students with equal bonuses**.

## Output

- Print the maximum bonus points along with the attendances of the given student, **rounded** to the nearest **bigger** number, scored by a student in this course in the format described above.

## Examples

Input	Output
5 25 30 12 19 24 16	Max Bonus: 34.  The student has attended 24 lectures.

20	
Comments	
First, we receive the <b>number of students</b> enrolled in the course – <b>5</b> . The total count of the lectures is <b>25</b> and the initial bonus is <b>30</b> . Then we calculate the bonus of the student with 12 attendances, which is <b>16.8</b> . We continue calculating <b>each of the student's bonuses</b> . The one <b>with 24 attendances</b> has the <b>highest bonus – 33.6 (34 rounded)</b> , so we print the appropriate message on the console.	
10 30 14 8 23 27 28 15 17 25 26 5 18	Max Bonus: 18.  The student has attended 28 lectures.

## 2. Shopping List

Link: <https://judge.softuni.bg/Contests/Practice/Index/2031#1>

*It's the end of the week and it is time for you to go shopping, so you need to create a shopping list first.*

### Input

You will receive an **initial list** with groceries separated by "!".

After that you will be receiving **4 types** of commands, until you receive **"Go Shopping!"**

- **Urgent {item}** - **add** the item at the **start** of the list. If the item **already exists**, skip this command.
- **Unnecessary {item}** - **remove** the item with the given name, only **if it exists** in the list. Otherwise skip this command.
- **Correct {oldItem} {newItem}** – if the item with the given **old name** exists, **change** its name with the **new** one. If it **doesn't exist**, skip this command.
- **Rearrange {item}** - if the grocery exists in the list, **remove** it from its **current position** and **add** it at the **end** of the list.

### Constraints

- There won't be any duplicate items in the initial list

## Output

Print the **list** with all the groceries, joined by ", ".

- "{firstGrocery}, {secondGrocery}, ...{nthGrocery}"

## Examples

Input	Output
Tomatoes!Potatoes!Bread Unnecessary Milk Urgent Tomatoes Go Shopping!	Tomatoes, Potatoes, Bread
Input	Output
Milk!Pepper!Salt!Water!Banana Urgent Salt Unnecessary Grapes Correct Pepper Onion Rearrange Grapes Correct Tomatoes Potatoes Go Shopping!	Milk, Onion, Salt, Water, Banana

## 3. Numbers

Link: <https://judge.softuni.bg/Contests/Practice/Index/2474#2>

Write a program to **read a sequence of integers** and find and print the **top 5** numbers that are **greater than the average** value in the sequence, sorted in descending order.

## Input

Read from the console a single line holding space separated number.

## Output

Print the above described numbers on a single line, space separated. If **less than 5 numbers** hold the above mentioned property, print less than 5 numbers. Print **"No"** if no numbers hold the above property.

## Constraints

All input numbers are integers in range [-1 000 000 ... 1 000 000]. The count of numbers is in range [1...10 000].

## Examples

Input	Output	Comments
10 20 30 40 50	50 40	Average number = 30. Numbers greater than 30 are: {40, 50}.

		<p>The top 5 numbers among them in descending order are: {50, 40}.</p> <p>Note that we have only 2 numbers, so all of them are included in the top 5.</p>
5 2 3 4 -10 30 40 50 20 50 60 60 51	60 60 51 50 50	<p>Average number = 28.08.</p> <p>Numbers greater than 28.08 are: {30, 40, 50, 50, 60, 60, 51}.</p> <p>The top 5 numbers among them in descending order are: {60, 60, 51, 50, 50}.</p>
1	No	<p>Average number = 1.</p> <p>There are no numbers, greater than 1.</p>
-1 -2 -3 -4 -5 -6	-1 -2 -3	<p>Average number = -3.5.</p> <p>Numbers greater than -3.5 are: {-1, -2, -3}.</p> <p>The top 5 numbers among them in descending order are: {-1, -2, -3}.</p>