# Mid Exam Preparation – 14 February 2024

## Black Flag

**Link:** [**https://judge.softuni.org/Contests/Practice/Index/1773#0**](https://judge.softuni.org/Contests/Practice/Index/1773#0)

*Pirates are invading the sea, and you're tasked to help them plunder*

Create a program that checks if **target plunder** is **reached**. First, you will receive how many **days** the pirating lasts. Then you will receive how much the pirates **plunder for a day**. Last you will receive the **expected plunder** at the end.

Calculate how much **plunder** the pirates manage to **gather**. Each **day** they gather the **plunder**. Keep in mind that they attack more ships every third day and add additional plunder to their total gain, which is **50% of the daily plunder**. Every **fifth day** the pirates encounter a warship, and after the battle, they **lose 30%** of their **total plunder**.

If the gained plunder is **more or equal** to the target, print the following:

**"Ahoy! {totalPlunder} plunder gained."**

If the gained plunder is **less** than the target. Calculate the **percentage left** and print the following:

**"Collected only {percentage}% of the plunder."**

Both numbers should be **formatted** to the **2nd decimal place**.

### Input

* On the **1st line,** you will receive the **days** of the plunder – an **integer number** in the range [0…100000].
* On the **2nd line,** you will receive the **daily plunder** – an **integer number** in the range [0…50].
* On the **3rd line,** you will receive the **expected plunder** – a **real number** in the range [0.0…10000.0].

### Output

* In the end, print whether the plunder **was successful** or **not,** following the format **described above**.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 40 100 | Ahoy! 154.00 plunder gained. |
| **Comments** | |
| The days are 5, and the daily plunder is 40. On the third day, the total plunder is 120, and since it is a third day, they gain an additional 50% from the daily plunder, which adds up to 140. On the fifth day, the plunder is 220, but they battle with a warship and lose 30% of the collected cargo, and the total becomes 154. That is more than expected. | |
|  | |
| 10  20  380 | Collected only 36.29% of the plunder. |

## Shoot for the Win

**Link:** [**https://judge.softuni.org/Contests/Practice/Index/2517#1**](https://judge.softuni.org/Contests/Practice/Index/2517#1)

Write a program that **finds a place for the tourist on a lift.**

Every wagon should have **a maximum of 4 people on it**. If a wagon is full, you should direct the people to **the next one with space** available.

### Input

* **On the first line,** you will receive **how many people** are waiting to get **on the lift**
* **On the second line**, you will receive the **current state of the lift separated by a single space:** **" "**.

### Output

**When there is no more available space left on the lift**, or there are **no more people in the queue**, you should print on the console the final state of the lift's wagons separated by **" "** and one of the following messages:

* If there are no more people and the lift have empty spots, you should print:

**"The lift has empty spots!**

**{wagons separated by ' '}"**

* If there are still people in the queue and no more available space, you should print:

**"There isn't enough space! {people} people in a queue!**

**{wagons separated by ' '}"**

* If the lift is full and there are no more people in the queue, you should print only the wagons separated by   
  **" ".**

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 15  0 0 0 0 | The lift has empty spots!  4 4 4 3 |
| **Comment** | |
| First state - 4 0 0 0 -> 11 people left  Second state – 4 4 0 0 -> 7 people left  Third state – 4 4 4 0 -> 3 people left | |
| **Input** | **Output** |
| 20  0 2 0 | There isn't enough space! 10 people in a queue!  4 4 4 |
| **Comment** | |
| First state - 4 2 0 -> 16 people left  Second state – 4 4 0 -> 14 people left  Third state – 4 4 4 -> 10 people left, but there're no more wagons. | |

## Numbers

**Link:** [**https://judge.softuni.org/Contests/Practice/Index/2474#2**](https://judge.softuni.org/Contests/Practice/Index/2474#2)

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

### Input

* Read from the console a single line holding **space-separated integers**.

### Output

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

### Constraints

* All input **numbers** are integers in the **range** [-1 000 000 … 1 000 000].
* The **count of numbers** is in the **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}.  The top 5 numbers among them in descending order are: {50, 40}.  Note that we have only 2 numbers, so all of them are included in the top 5. |
| 5 2 3 4 -10 30 40 50 20 50 60 60 51 | 60 60 51 50 50 | Average number = 28.08.  Numbers greater than 28.08 are: {30, 40, 50, 50, 60, 60, 51}.  The top 5 numbers among them in descending order are: {60, 60, 51, 50, 50}. |
| 1 | No | Average number = 1.  There are no numbers greater than 1. |
| -1 -2 -3 -4 -5 -6 | -1 -2 -3 | Average number = -3.5.  Numbers greater than -3.5 are: {-1, -2, -3}.  The top 5 numbers among them in descending order are: {-1, -2, -3}. |