**Exercise: Lists Basics**

Problems for exercise and homework for the [Python Fundamentals Course @SoftUni](https://softuni.bg/trainings/3003/python-fundamentals-september-2020). Submit your solutions in the SoftUni judge system at <https://judge.softuni.bg/Contests/1725>

* **Invert Values**

Write a program that receives a **single string** containing **numbers** separated by a **single space**. Print a list containing the **opposite of each number.**

**Example**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 2 -3 -3 5 | [-1, -2, 3, 3, -5] |
| -4 0 2 57 -101 | [4, 0, -2, -57, 101] |

* **Multiples List**

Write a program that receives **two numbers** (factor and count) and creates a **list** with **length** of the given **count** and contains only **elements** that are **multiples** of the given **factor.**

**Example**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  5 | [2, 4, 6, 8, 10] |
| 1  10 | [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] |

* **Football Cards**

*Most football fans love it for the goals and excitement. Well, this problem doesn't. You are to handle the referee's little notebook and count the players who were sent off for fouls and misbehavior.*

The rules: **Two teams**, named "**A**" and "**B**" have **11 players** each. The players on each team are **numbered** from **1 to 11**. Any player may be **sent off** the field by being given a **red card**. If one of the teams has **less than 7 players** remaining, the game is **stopped** immediately by the referee, and the **team with less than 7 players loses**.

A **card** is a **string** with the **team's letter** (**'A' or 'B'**) followed by a **single dash** and **player's number**. **e.g.** the card **'B-7'** means player **#7** from team **B** received a **card**.

The task: Given a list of cards (could be empty), return the number of **remaining players** on **each team** at the **end of the game** in the format: **"Team A - {players\_count}; Team B - {players\_count}".** If the game was terminated print an additional line: **"Game was terminated"**

***Note for the random tests***: If a player that has already been sent off receives another card - ignore it.

**Input**

The input (the cards) will come on a **single line** separated by a **single space.**

**Output**

Print the remaining players as described above and add another line (as shown above) if the game was terminated.

**Example**

|  |  |
| --- | --- |
| **Input** | **Output** |
| A-1 A-5 A-10 B-2 | Team A - 8; Team B - 10 |
| A-1 A-5 A-10 B-2 A-10 A-7 A-3 | Team A - 6; Team B - 10  Game was terminated |

* **Number Beggars**

Your task here is pretty simple: given a **list of numbers** and a **number of beggars**, you are supposed to return a **list with the sum** of what **each beggar** brings home, assuming they all take **regular turns**, from the first to the last.

For example: **[1,2,3,4,5]** for 2 beggars will return a result of **9 and 6**, as the first one takes **[1,3,5]**, the second collects **[2,4]**.

The same list with **3 beggars** would produce a better outcome for the **second** beggar: **5, 7 and 3**, as they will respectively take **[1, 4], [2, 5] and [3]**.

Also note that not all beggars have to take the same amount of "offers", meaning that the length of the list is **not** necessarily a **multiple of n**; length can be even shorter, in which case the last beggars will of course take nothing (0).

**Input**

You will receive **2 lines** of input: a **single string** containing the numbers separated by a comma and a space **", "**. On the **second line** you will receive the **number of beggars.**

**Output**

Print a **list** of all the **sums** that each beggar got.

**Example**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1, 2, 3, 4, 5  2 | [9, 6] |
| 3, 4, 5, 1, 29, 4  6 | [3, 4, 5, 1, 29, 4] |
| 100, 94, 24, 99  5 | [100, 94, 24, 99, 0] |

* **Faro Shuffle**

A faro shuffle of a deck of playing cards is a shuffle in which the deck is **split exactly in half** and then the cards in the two halves are **perfectly interwoven**, such that the **original bottom card is still on the bottom and the original top card is still on top**.

For example, faro shuffling the list

**['ace', 'two', 'three', 'four', 'five', 'six']** once, gives **['ace', 'four', 'two', 'five', 'three', 'six']**

Write a program that receives a **single string** (cards separated by **space**) and on the **second line** receives a **number** of faro **shuffles** that have to be made. Print the **state of the deck after the shuffle.**

***Note*: The length of the deck of cards will always be an even number**

**Example**

|  |  |
| --- | --- |
| **Input** | **Output** |
| a b c d e f g h  5 | ['a', 'c', 'e', 'g', 'b', 'd', 'f', 'h'] |
| one two three four  3 | ['one', 'three', 'two', 'four'] |

* **Survival of the Biggest**

Write a program that receives a **list of integer** numbers and a number **n**. The number **n** represents the **amount of numbers to remove** from the list. You should remove the **smallest ones.**

**Input**

On the first line you will receive a string (numbers separated by a space), on the second line you will receive a number n (count of numbers to remove).

**Output**

Print all the numbers that are left in the list.

**Example**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 10 9 8 7 6 5  3 | [10, 9, 8] |
| 1 10 2 9 3 8  2 | [10, 9, 3, 8] |