# The Hunting Games

*A group of friends have decided to participate in a game called "The Hunting Games". The first stage of the game is to gather some supplies. They have a list and your job is to help them follow it and make the needed calculations.*

Write a program that calculates the needed provisions for a quest in the woods.

First you will receive **the days of the adventure**, **the count of the players** and the **group’s energy**. Afterwards, you will receive the following **provisions per day for one person**:

* **Water**
* **Food**

**Every day** they chop wood and **lose** **a certain amount of energy**. For each of the days, you are going to receive the energy loss from chopping wood. The program should **end** If the energy reaches **0** or **less**.

**Every second day** they **drink water**, which **boosts** their **energy** with **5% of their current energy** and at the same **time drops their water** **supplies** by **30% of their current water.**

**Every** **third** day they **eat**, which **reduces their food supplies by the following amount:**

{currentFood} / {countOfPeople} and at the same time **raises** their group’s **energy** by **10%**.

If they have **enough** **energy** to finish the quest, print the following message:

"You are ready for the quest. You will be left with - {energyLevel} energy!"

If they **run out of energy** print the following message and the **food** and **water** they were left with **before** they ran out of energy:

"You will run out of energy. You will be left with {food} food and {water} water."

## Input / Constraints

* **On the 1st line**, you are going to receive the days of the adventure – **an integer** in the range **[1…100]**
* **On the 2nd line** – the count of players – **an integer** in the range **[0 – 1000]**
* **On the 3rd line** - the group’s energy – **a real number** in the range **[1 - 50000]**
* **On the 4th line** – water per day for one person – **a real number** **[0.00 – 1000.00]**
* **On the 5th line** – food per day for one person – **a real number** **[0.00 – 1000.00]**
* On the next **n** lines – one for each of the days – the amount of **energy loss**– **a real number** in the range **[0.00 - 1000]**
* You will **always** have **enough** **food** and **water**.

## Output

* "You are ready for the quest. You will be left with - {energyLevel} energy!" –   
  if they have enough energy

"You will run out of energy. You will be left with {food} food and {water} water."

* All of the real numbers should be **formatted** to the **second** **digit** after the decimal separator

## Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 10  7  5035.5  11.3  7.2  942.3  500.57  520.68  540.87  505.99  630.3  784.20  321.21  456.8  330 | You are ready for the quest. You will be left with - 658.72 energy! |
| **Comments** | |
| The **days** are **10** and the **players** are **7**. The **energy** of the whole **group** is **5035.5**. We receive the **water** and **food** and we can **calculate** the needed amount of both for the whole quest:  **10 \* 7 \* 11.3 – total water = 791**  **10 \* 7 \* 7.2 – total food = 504**  Afterwards, for **each** of the **days** you have to calculate the energy loss. On each day you receive energy loss and you have to **subtract** it. On the first day it is:  **5035.5 – 942.3 = 4093.2**  On **every second day** we **add** the **energy boost** from the drank water, which is **5% of the current energy** and **subtract** the amount **from the total water**. The first time we reach a second day, the energy will become **3772.26** and the water will become **553.7**. The first time we reach a **third** **day**, we have to **boost the energy with 10%** and **reduce** the food supplies and the energy will become - **3576.74** and the food **432**. Make all of the calculations and in the end, you must have **658.77** energy left and **132.94** water and **317.39** food left. | |

|  |  |
| --- | --- |
| **Input** | **Output** |
| 12  6  4430  9.8  5.5  620.3  840.2  960.1  220  340  674  365  345.5  212  412.12  258  496 | You will run out of energy. You will be left with 229.17 food and 118.59 water. |

# Seize the Fire

*The group of adventurists have gone on their first task. Now they have to walk through fire - literally. They have to use all of the water they have left. Your task is to help them survive.*

Create a program that calculates the water that is needed to put out a "fire cell", based on the given information about its "fire level" and how much it gets affected by water.

First, you will be given **the level of fire** inside the cell with the **integer** **value** of the **cell**, which represents the needed water to put out the fire. They will be given in the following format:

**"{typeOfFire} = {valueOfCell}#{typeOfFire} = {valueOfCell}#{typeOfFire} = {valueOfCell}……**"

Afterwards you will receive the **amount of water** you have for putting out the fires. There is a **range** of fire for each of the fire types, and if a cell’s value is below or exceeds it, it is invalid and you don’t need to put it out.

|  |  |
| --- | --- |
| **Type of Fire** | **Range** |
| High | 81 - 125 |
| Medium | 51 - 80 |
| Low | 1 - 50 |

If a cell is valid, you have to put it out by reducing the water with its value. Putting out fire also takes **effort** and you need to **calculate it**. Its value is equal to **25% of the cell’s value**. In the end you will have to print the **total effort**. Keep putting out cells until you run out of water. If you **don’t have enough** **water** to put out a given cell – **skip it** and **try the next one**. In the end, **print the cells you have put out** in the following format:

**"Cells:**

**- {cell1}**

**- {cell2}**

**- {cell3}**

**……**

**- {cellN}"**

**"Effort: {effort}"**

In the end, print the total fire you have put out from all of the cells in the following format: "Total Fire: {totalFire}"

## Input / Constraints

* **On the 1st line** you are going to receive the **fires with their cells** in the format described above **– integer numbers in the range [1…500]**
* **On the 2nd line**, you are going to be given the **water** – **an integer number** in the range **[0….100000]**

## Output

* Print the cells, which you have put out in the following format:

"Cells:

- {cell}

- {cell2}

- {cell3}

- {cell5}

……

- {cellN}"

* Print the effort, rounded 2 digits after the decimal separator in the following format:

"Effort: {effort}"

* Print the total fire put out

"Total Fire: {totalFire}"

## Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| High = 89#Meduim = 53#Low = 28#Medium = 77#Low = 23  1250 | Cells:  - 89  - 28  - 77  - 23  Effort: 54.25  Total Fire: 217 |
| **Comments** | |
| After reading the output, we start **checking** the **level of the fire** and its validity. The first is valid, so we **subtract the 89** from the amount of **water** – 1250, and the water becomes 1161. We need to calculate the **effort**, which is **25%** of 89. We will **add 89 to the total fire** we have put out. In the end the effort is 54.22 and the total fire: 217 | |

|  |  |
| --- | --- |
| **Input** | **Output** |
| High = 150#Low = 55#Medium = 86#Low = 40#High = 110#Medium = 77  220 | Cells:  - 40  - 110  Effort: 37.50  Total Fire: 150 |

# The Final Quest

*After walking through fire, the group has reached the final step of the quest. They have received a list with instructions that will help them resolve the last riddle that will lead them to the truth about the Hunting Games.*

Create a program that **follows** given **instructions**. You will receive a **collection of words on a single line**, split by a **single space**. They are not what they are supposed to be, so you have to **follow the instructions** in order to find the **real message**. You will be receiving commands. Here are the possible ones:

* Delete {index} – removes the word **after** the given index **if it is valid**.
* Swap {word1} {word2} – find the given words in the collections **if they exist** and **swap** their places.
* Put {word} {index} – **add** a word at the **previous place {index} before** the   
  given one, **if it exists**.
* Sort – you must **sort** the words in descending order.
* Replace {word1} {word2}– find the **second word** **{word2}** in the collection (**if it exists**) and **replace** it with the **first word – {word1}**.

Follow them until you receive the "**Stop**" command. After you have successfully **followed the instructions**, you must print the words on a single line, split by a space.

## Input / Constraints

* **On the 1st line**, you are going to receive the collection of words, split by a single space – **strings**
* **On the next lines**, you are going to receive **commands**, until you receive the "**Stop**" command

## Output

* Print the words you have gathered on a single line, split by a single space

## Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Congratulations! You last also through the have challenge!  Swap have last  Replace made have  Delete 2  Put it 4  Stop | Congratulations! You made it through the last challenge! |
| **Comments** | |
| First, we receive the command **“Swap”,** so we change the positions of the words **have** and **last**. The text at this point should look like this:  **Congratulations! You have also through the last challenge!**  After that, we receive **“Replace”** and we have to replace the **second word** – “have” **with the first** – “made”. Afterwards we have to **delete** the **word**, which is **after the second index**. And finally, we have to put a word on the **previous** position **before** **4**. | |

|  |  |
| --- | --- |
| **Input** | **Output** |
| This the my quest! final  Put is 2  Swap final quest!  Delete 2  Stop | This is the final quest! |