# 02. Collecting Stars Game



*In this thrilling adventure, you are a star collector navigating a mysterious, obstacle-filled field, searching for ten elusive stars. Use your wits and skillful movements to gather stars, avoid obstacles, and achieve victory before your stars run out!*

On the **first line**, you will be given an **integer N**, **representing the size** of the **field** with a **square shape**.

On the **following N lines**, you will be given the **field** containing symbols, separated by a single space. See the [**Examples**](#_Examples) section.

Your **goal** is to **collect 10 stars**. **10**

The **field will contain randomly positioned elements** - a **player**, **stars**, and **obstacles**:

* **One player**, **marked** with the **letter "P"**
* **Stars**, **marked** with the **asterisk** **symbol "\*"**
* **Obstacles** **marked** with **hashtag symbol** **"#"**

There are **two possible outcomes** of the **game** and **commands** are received until:

* The player **collects** **10 stars** and **wins** the **game**.
* The player **hits obstacles**, **loses all his stars**, and **loses** the **game**.

After the field state, you will be **given commands** for the **player's movement**. **Commands** can be: **"up"**, **"down"**, **"left"**, or **"right"**.

The **player** **starts** the **game with 2 stars initially** **2** and **moves** in the **given direction** with **one step for each command**, **collecting all the stars he comes across or losing a star each time he hits an obstacle**.

The player **can go through** the **same path many times** but **can collect the stars just once** (the first time), while the **obstacles** are **immovable** and **will remain** there. The player **can hit the same obstacle** **many times** and **lose** a **star** **each time** that happens.

**Game rules**:

* When the **player comes across a star** and **collects** **it**, the cell shall be **marked with a dot "."**.
  + The **total number of collected stars** shall be **increased by one**. **+1**
* If the **player encounters an obstacle**, he **does not move** and **remains** in **his current position**.
  + The **player loses one star each time he hits an obstacle**. **-1**
* If the **player** **steps out of the field**, he will be punished by a **teleport** to the **field's starting position** ( coordinates [**0, 0**]).
* When the **player makes his first move**, mark his **initial position** with a **dot** **"."**.
* The **game** **continues until** the player manages to **collect 10 stars** and **wins** **or** **until** he **loses all his stars** by hitting obstaclesand therefore **loses the game**. See the [**Examples**](#_Examples) section.
* **At** the **end of the game**, **print** the **final state** of the **field** and **the player's final position**, **marked with "P"**.

## Input

* On the **first line** you will receive an **integer** **N** representing the **size of the square field** (matrix **NxN**).
* On the **next N lines** you will get the **field** **rows** (**each position separated by a single space**)
* On **each of the following lines**, you will get a **valid** **move command**.

## Output

At the **end** of the program:

* If the player **won the game**, print: **"You won! You have collected 10 stars."**
* If the player **loses the game**, print: **"Game over! You are out of any stars."**
* Next, print the **player's final position**: **"Your final position is [{row\_position}, {column\_position}]"**
* Finally, print the **matrix in its final state**, **each position separated by a single space**. Remember to **mark the player's final position with "P"**.

## Constraints

* There will always be **enough commands** to **either win or lose the game**.
* There will be **no case** inwhich **less than 10 stars will be in the field**.
* There will be **no obstacle** at the **field's** **starting position** (coordinates [0,0])
* All **given symbols** will be **valid** following the **description**.

**See the Examples section below**

## Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5  \* # \* \* \*  # \* \* \* \*  \* \* \* \* #  P \* \* \* \*  \* \* \* # \*  right  right  right  up  up  up  right  down  left  left  left  down | You won! You have collected 10 stars.  Your final position is [1, 4]  \* # \* . .  # \* \* . P  \* \* \* . #  . . . . \*  \* \* \* # \* |
| **Comment** | |
| The **program starts** with the **player placed at coordinates [3, 0]** and **having 2 stars initially**.  The commands are processed as follows:  right: Moves from [3, 0] to [3, 1], collects one star and now has **3** stars in total  right: Moves from [3, 1] to [3, 2], collects one star and now has **4** stars in total  right: Moves from [3, 2] to [3, 3], collects one star and now has **5** stars in total  up: Moves from [3, 3] to [2, 3], collects one star, and now has **6** stars in total  up: Moves from [2, 3] to [1, 3], collects one star, and now has **7** stars in total  up: Moves from [1, 3] to [0, 3], collects one star, and now has **8** stars in total  right: Moves from [0, 3] to [0, 4], collects one star and now has **9** stars in total  down: Moves from [0, 4] to [1, 4], collects one star and now has **10** stars in total  **Win**: The **player collected 10 stars** and **finished the game at [1, 4]**. The player's **path** was **marked with dots** as he **collected stars** on **each move**.  The **appropriate messages** are **printed** indicating the **success**. | |
| **Input** | **Output** |
| 4  \* # \* \*  # P # \*  \* \* \* \*  \* \* \* \*  up  down  up  up  up  up  down  left  left  left  down | Game over! You are out of any stars.  Your final position is [1, 1]  \* # \* \*  # P # \*  \* . \* \*  \* \* \* \* |
| **Input** | **Output** |
| 4  \* \* # \*  \* \* # \*  P \* # \*  # # \* \*  right  up  left  left  down  up  right  right  right  right  right  up  right  right  right  right  right | Game over! You are out of any stars.  Your final position is [0, 1]  . P # \*  . . # \*  . . # \*  # # \* \* |