# Programming Fundamentals with Python: Exam Preparation

## 01. Activation Keys

**Submit your solutions in the SoftUni judge system at https://judge.softuni.org/Contests/Practice/Index/2302#0.**

*You are about to make some good money, but first, you need to think of a way to verify who paid for your product and who didn't. You have decided to let people use the software for a free trial period and then require an activation key to continue using the product. Before you can cash out, the last step is to design a program that creates unique activation keys for each user. So, waste no more time and start typing!*

The first line of the input will be your raw activation key. It will consist of **letters and numbers only**.

After that, until the **"Generate"** command is given, you will be receiving strings with instructions for different operations that need to be performed upon the raw activation key.

There are several types of instructions, split by **">>>"**:

- **"Contains>>>{substring}"**:
    - If the raw activation key contains the given substring, prints: **"{raw activation key} contains {substring}"**.
    - Otherwise, prints: **"Substring not found!"**
- **"Flip>>>Upper/Lower>>>{startIndex}>>>{endIndex}"**:
    - Changes the substring **between the given indices (the end index is exclusive)** to upper or lower case and then prints the activation key.
    - All given indexes will be valid.
- **"Slice>>>{startIndex}>>>{endIndex}"**:
    - **Deletes** the characters between the start and end indices (**the end index is exclusive) and** prints the activation key.
    - Both indices will be **valid**.

### Input

- The first line of the input will be a string consisting of **letters and numbers only**.
- After the first line, until the **"Generate"** command is given, you will be receiving **strings**.

### Output

- After the **"Generate"** command is received, print:
    - **"Your activation key is: {activation key}"**

### Examples

| Input | Output |
|---|---|
| abcdefghijklmnopqrstuvwxyz | abghijklmnopqrstuvwxyz |
| Slice>>>2>>>6 | abgHIJKLMNOPQRstuvwxyz |
| Flip>>>Upper>>>3>>>14 | abgHIjkLMNOPQRstuvwxyz |
| Flip>>>Lower>>>5>>>7 | Substring not found! |

| | |
|---|---|
| Contains>>>def<br><br>Contains>>>deF<br><br>Generate | Substring not found!<br><br>Your activation key is: abgHIjkLMNOPQRstuvwxyz |

| Comments |
|---|
| 1. **Slice>>2>>6**<br><br>ab<mark style="background:red">cdef</mark>ghijklmnopqrstuvwxyz becomes abghijklmnopqrstuvwxyz<br><br>2. **Flip>>>Upper>>>3>>>14**<br><br>abg<mark style="background:yellow">hijklmnopqr</mark>stuvwxyz becomes abg<mark style="background:yellow">HIJKLMNOPQR</mark>stuvwxyz<br><br>3. **Flip>>>Lower>>>5>>>7**<br><br>abgHI<mark style="background:lime">JK</mark>LMNOPQRstuvwxyz becomes abgHI<mark style="background:lime">jk</mark>LMNOPQRstuvwxyz<br><br>4. **Contains>>>def**<br><br>abgHIjkLMNOPQRstuvwxyz does not contain def<br><br>5. **Contains>>>deF**<br><br>abgHIjkLMNOPQRstuvwxyz does not contain deF<br><br>The final activation key is abgHIjkLMNOPQRstuvwxyz |

| Input | Output |
|---|---|
| 134softsf5ftuni2020rockz42<br><br>Slice>>>3>>>7<br><br>Contains>>>-rock<br><br>Contains>>>-uni-<br><br>Contains>>>-rocks<br><br>Flip>>>Upper>>>2>>>8<br><br>Flip>>>Lower>>>5>>>11<br><br>Generate | 134sf5ftuni2020rockz42<br><br>Substring not found!<br><br>Substring not found!<br><br>Substring not found!<br><br>134SF5FTuni2020rockz42<br><br>134SF5ftuni2020rockz42<br><br>Your activation key is: 134SF5ftuni2020rockz42 |

## 02. Mirror words

**Submit your solutions in the SoftUni judge system at https://judge.softuni.org/Contests/Practice/Index/2307#1.**

*The SoftUni Spelling Bee competition is here. But it`s not like any other Spelling Bee competition out there. It`s different and a lot more fun! You, of course, are a participant, and you are eager to show the competition that you are the best, so go ahead, learn the rules and win!*

On the first line of the input, you will be given a **text string**. To win the competition, you have to find all hidden **word pairs**, read them, and mark the ones that are **mirror images** of each other.

First of all, you have to **extract the hidden word pairs**. Hidden word pairs are:

- Surrounded by **"@"** or **"#"** (only one of the two) in the following pattern **#wordOne##wordTwo#** or **@wordOne@@wordTwo@**
- At least **3 characters long each** (**without the surrounding symbols**)
- Made up of **letters only**

If the second word, **spelled backward,** is the **same as the first word and vice versa** (**casing matters**!), they are a **match,** and you have to store them somewhere. **Examples** of mirror words:

**#Part##traP# @leveL@@Level@ #sAw##wAs#**

- If you don`t find any valid pairs, print: **"No word pairs found!"**
- If you find valid pairs print their count: **"{valid pairs count} word pairs found!"**
- If there are no mirror words, print: **"No mirror words!"**
- If there are mirror words print:

  **"The mirror words are:**

  **{wordOne} <=> {wordtwo}, {wordOne} <=> {wordtwo}, … {wordOne} <=> {wordtwo}"**

## Input / Constraints

- You will recive a string.

## Output

- Print the proper output messages in the proper cases as described in the problem description.
- If there are pairs of mirror words, print them in the end, each pair separated by **", "**.
- Each pair of mirror word must be printed with **" <=> "** between the words.

## Examples

| Input |
|---|
| @mix#tix3dj#poOl##loOp#wl@@bong&song%4very$long@thong#Part##traP##@@leveL@@Level@#car#rac##tu@pack@@ckap@#rr#sAw##wAs#r#@w1r |

| Output | Comments |
|---|---|
| 5 word pairs found!<br><br>The mirror words are:<br>Part <=> traP, leveL <=> Level, sAw <=> wAs | There are 5 green and yellow pairs that meet all requirements and thus are valid.<br><br>#poOl##loOp# is valid and looks very much like a mirror words pair, but it isn`t because the casings don`t match.<br><br>#car#rac# "rac" spelled backward is "car", but this is not a valid pair because there is only one "#" between the words.<br><br>@pack@@ckap@ is also valid, but "ckap" backward is "pakc" which is not the same as "pack", so they are not mirror words. |

| Input |
|---|
| #po0l##l0op# @bAc##cAB@ @LM@ML@ #xxxXxx##xxxXxx# @aba@@ababa@ |

Follow us:

| Output | Comments |
|---|---|
| 2 word pairs found!<br><br>No mirror words! | "xxxXxx" backward is not the same as "xxxXxx"<br><br>@aba@@ababa@ is a valid pair, but the word lengths are different - these are definitely not mirror words |
| **Input** | |
| #lol#lol# @#God@@doG@# #abC@@Cba# @Xyu@#uyX# | |
| **Output** | **Comments** |
| No word pairs found!<br><br>No mirror words! | |

# 03. Heroes of Code and Logic VII

**Submit your solutions in the SoftUni judge system at https://judge.softuni.org/Contests/Practice/Index/2303#2.**

*You got your hands on the most recent update on the best MMORPG of all time – Heroes of Code and Logic. You want to play it all day long! So cancel all other arrangements and create your party!*

On the first line of the standard input, you will receive an integer **n** – the number of heroes that you can choose for your party. On the next **n** lines, the heroes themselves will follow with their **hit points** and **mana points** separated by a single space in the following format:

**"{hero name} {HP} {MP}"**

- **HP stands for hit points and MP for mana points**
- **a hero can have a maximum of 100 HP and 200 MP**

After you have successfully picked your heroes, you can start playing the game. You will be receiving different commands, each on a new line, separated by **" – ",** until the **"End"** command is given.

There are several actions that the heroes can perform:

**"CastSpell – {hero name} – {MP needed} – {spell name}"**
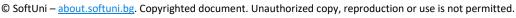
- If the hero has the required MP, he casts the spell, thus reducing his MP. Print this message:
  - **"{hero name} has successfully cast {spell name} and now has {mana points left} MP!"**

- If the hero is unable to cast the spell print:
  - **"{hero name} does not have enough MP to cast {spell name}!"**

**"TakeDamage – {hero name} – {damage} – {attacker}"**

- Reduce the hero HP by the given damage amount. If the hero is still alive (his HP is greater than 0) print:
  - **"{hero name} was hit for {damage} HP by {attacker} and now has {current HP} HP left!"**
- If the hero has died, remove him from your party and print:
  - **"{hero name} has been killed by {attacker}!"**

**"Recharge – {hero name} – {amount}"**

- The hero increases his MP. If it brings the MP of the hero above the **maximum value** (**200**), MP is increased to **200**. (the MP can't go over the maximum value).
- Print the following message:
  - **"{hero name} recharged for {amount recovered} MP!"**

**"Heal – {hero name} – {amount}"**

- The hero increases his HP. If a command is given that would bring the HP of the hero above the **maximum value (100)**, HP is increased to **100** (the HP can't go over the maximum value).
- Print the following message:
  - **"{hero name} healed for {amount recovered} HP!"**

## Input

- On the first line of the standard input, you will receive an integer **n**
- On the following **n** lines, the heroes themselves will follow with their **hit points** and **mana points** separated by a space in the following format
- You will be receiving different **commands**, each on a new line, separated by **" – "**, until the **"End"** command is given

## Output

- Print all members of your party who are **still alive**, in the following format (their HP/MP need to be indented 2 spaces):

**"{hero name}**

　**HP: {current HP}**

　**MP: {current MP}"**

## Constraints

- The starting HP/MP of the heroes will be valid, 32-bit integers will never be negative or exceed the respective limits.
- The HP/MP amounts in the commands will never be negative.
- The hero names in the commands will always be valid members of your party. No need to check that explicitly.

## Examples

| Input | Output |
|-------|--------|
| 2<br><br>Solmyr 85 120<br><br>Kyrre 99 50<br><br>Heal - Solmyr - 10<br><br>Recharge - Solmyr - 50<br><br>TakeDamage - Kyrre - 66 - Orc<br><br>CastSpell - Kyrre - 15 - ViewEarth | Solmyr healed for 10 HP!<br><br>Solmyr recharged for 50 MP!<br><br>Kyrre was hit for 66 HP by Orc and now has 33 HP left!<br><br>Kyrre has successfully cast ViewEarth and now has 35 MP!<br><br>Solmyr<br><br>　HP: 95 |

| | |
|---|---|
| End | MP: 170 |
| | Kyrre |
| | HP: 33 |
| | MP: 35 |

| Input | Output |
|---|---|
| 4 | SirMullich healed for 30 HP! |
| Adela 90 150 | Adela recharged for 50 MP! |
| SirMullich 70 40 | Tyris does not have enough MP to cast Fireball! |
| Ivor 1 111 | Tyris has been killed by Fireball! |
| Tyris 94 61 | Ivor has been killed by Mosquito! |
| Heal - SirMullich - 50 | Adela |
| Recharge - Adela - 100 | HP: 90 |
| CastSpell - Tyris - 1000 - Fireball | MP: 200 |
| | SirMullich |
| TakeDamage - Tyris - 99 - Fireball | HP: 100 |
| TakeDamage - Ivor - 3 - Mosquito | MP: 40 |
| End | |

| Comments |
|---|

Heal – SirMullich healed for 30 HP due to the HP max limit.

Recharge – Adela recharged for 50 MP due to the MP max limit.

CastSpell – Tyris does not have enough MP to cast the spell.

TakeDamage – Tyris's HP is reduced by 99, thus becoming -5, which means he is dead.

TakeDamage – Ivor's HP is now -2, so he is dead too.

After the "End" command, we print the remaining living heroes.