# Problem 1. Party Profit

*As a young adventurer, you travel with your party around the world, seeking for gold and glory. But you need to split the profit among your companions.*

You will receive a **party size**. After that you receive the **days** of the adventure.

**Every day,** you are **earning 50 coins**, but you also spent **2 coin per companion** for food.

Every **3rd (third)** day, you have a motivational party, spending **3 coins per companion** for drinking water.

Every **5th (fifth)** day you slay a boss monster and you **gain 20 coins per companion**. But if you have a motivational party the same day, you **spent additional 2 coins per companion**.

Every **10th (tenth)** day **at the start of the day**, **2 (two)** of your companions **leave**, but every **15th (fifteenth) day 5 (five)** **new** companions are joined **at the beginning of the day**.

You have to calculate how much coins gets each companion at the end of the adventure.

## Input / Constraints

The input will consist of **exactly 2 lines**:

* party size – **integer in range [1…100]**
* days **– integer in range [1…100]**

## Output

Print the following message: "{companionsCount} companions received {coins} coins each."

You cannot split a coin, so take the integral part (round down the coins to integer number).

## Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  5 | 3 companions received 90 coins each. |
| **Input** | **Output** |
| 15  30 | 19 companions received 102 coins each. |

# Problem 2. Dungeonest Dark

*As a young adventurer, you seek gold and glory in the darkest dungeons there are.*

You have **initial health 100 and initial coins 0**. You will be given **a string, representing the dungeons rooms**. Each room is separated with **'|'** (vertical bar): **"room1|room2|room3…"**

Each room contains item or a monster and a number, separated by space. (**"item/monster number"**)

* If the first part is **"potion":** you are healed with the number in the second part. But your health **cannot exceed** your **initial health (100)**. Print: **"You healed for {0} hp."**.

After that, print your current health: **"Current health: {0} hp."**.

* If the first part is **"chest"**: You've found some coins, the number in the second part. Print: **"You found {0} coins."**.
* In any other case you are facing a monster, you are going to fight. The second part of the room, contains the attack of the monster. You should remove the monster's attack from your health.
  + If you are not dead (health <= 0) you've slain the monster, and you should print (**"You slayed {monster}."**)
  + If you've died, print **"You died! Killed by {monster}."** and your quest is over. Print the best room you`ve manage to reach: **"Best room: {room}"**.

If you managed to go trough all the rooms in the dungeon, print on the next three lines:

**"You've made it!"**, **"Coins: {coins}"**, **"Health: {health}"**.

## Input / Constraints

You receive a string, representing the dungeons rooms, separated with **'|'** (vertical bar): **"room1|room2|room3…"**.

## Output

Print the corresponding messages, described above.

## Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| rat 10|bat 20|potion 10|rat 10|chest 100|boss 70|chest 1000 | You slayed rat.  You slayed bat.  You healed for 10 hp.  Current health: 80 hp.  You slayed rat.  You found 100 coins.  You died! Killed by boss.  Best room: 6 |
| **Input** | **Output** |
| cat 10|potion 30|orc 10|chest 10|snake 25|chest 110 | You slayed cat.  You healed for 10 hp.  Current health: 100 hp.  You slayed orc.  You found 10 coins.  You slayed snake.  You found 110 coins.  You've made it!  Coins: 120  Health: 65 |

# Problem 3. Quests Journal

*As a young adventurer, you start new quest every day, until you retire.*

## Input / Constraints

You start your adventurer path, receiving a journal with some beginner quests, separated with **', '** (comma and space). After that, until receiving "Retire!" you will be receiving different commands.

Commands:

* "Start - {quest}" – Receiving this command, you should add the given quest in your journal. If the quest already **exists**, you should **skip** this line.
* "Complete - {quest}" – You should remove the quest from your journal, **if it exists**.
* "Side Quest - {quest}:{sideQuest}" – You should check **if the quest exists**, if so, **add** the side quest **after** the quest. Note that you **shouldn`t add** the **sideQuest** if it already exists.
* "Renew – {quest}" – If the given quest exists, you should change its position and **put it last** in your journal.

## Output

After receiving "Retire!" print the quests in the journal, separated by **", "** (comma and space).

## Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Hello World, For loop, If else  Start - While loop  Complete - For loop  Retire! | Hello World, If else, While loop |
| **Input** | **Output** |
| Hello World, If else  Complete - Homework  Side Quest - If else:Switch  Renew - Hello World  Retire! | If else, Switch, Hello World |