# Problem 1 - Black Flag

Problem for exam preparation for the [Programming Fundamentals Course @SoftUni](https://softuni.bg/trainings/3447/programming-fundamentals-with-csharp-september-2021).

Submit your solutions in the SoftUni judge system at [https://judge.softuni.org/Contests/Practice/Index/1773#0](https://judge.softuni.org/Contests/Practice/Index/1773" \l "0).

*Pirates are invading the sea, and you're tasked to help them plunder*

Create a program that checks if **target plunder** is **reached**. First, you will receive how many **days** the pirating lasts. Then you will receive how much the pirates **plunder for a day**. Last you will receive the **expected plunder** at the end.

Calculate how much **plunder** the pirates manage to **gather**. Each **day** they gather the **plunder**. Keep in mind that they attack more ships every third day and add additional plunder to their total gain, which is **50% of the daily plunder**. Every **fifth day** the pirates encounter a warship, and after the battle, they **lose 30%** of their **total plunder**.

If the gained plunder is **more or equal** to the target, print the following:

**"Ahoy! {totalPlunder} plunder gained."**

If the gained plunder is **less** than the target. Calculate the **percentage left** and print the following:

**"Collected only {percentage}% of the plunder."**

Both numbers should be **formatted** to the **2nd decimal place**.

## Input

* On the **1st line,** you will receive the **days** of the plunder – an **integer number** in the range [0…100000]
* On the **2nd line,** you will receive the **daily plunder** – an **integer number** in the range [0…50]
* On the **3rd line,** you will receive the **expected plunder** – a **real number** in the range [0.0…10000.0]

## Output

* In the end, print whether the plunder **was successful** or **not,** following the format **described above**.

## Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 40 100 | Ahoy! 154.00 plunder gained. |
| **Comments** | |
| The days are 5, and the daily plunder is 40. On the third day, the total plunder is 120, and since it is a third day, they gain an additional 50% from the daily plunder, which adds up to 140. On the fifth day, the plunder is 220, but they battle with a warship and lose 30% of the collected cargo, and the total becomes 154. That is more than expected. | |
|  | |
| 10  20  380 | Collected only 36.29% of the plunder. |

# Problem 2. Mu Online

Problem for exam preparation for the [Programming Fundamentals Course @SoftUni](https://softuni.bg/trainings/3447/programming-fundamentals-with-csharp-september-2021).

Submit your solutions in the SoftUni judge system at [https://judge.softuni.org/Contests/Practice/Index/2028#1](https://judge.softuni.org/Contests/Practice/Index/2028" \l "1).

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

Each room contains **a command** and a **number**, separated by space. The command can be:

* **"potion"**
  + You are healed with the number in the second part. But your health **cannot exceed** your **initial health (100)**.
  + First print: **"You healed for {amount} hp."**
  + After that, print your current health: **"Current health: {health} hp."**
* **"chest"**
  + You've found some bitcoins, the number in the second part.
  + Print: **"You found {amount} bitcoins."**
* In **any other case,** you are **facing a monster**, which you will **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 through all the rooms** in the dungeon, print on the **following three lines**:

**"You've made it!"**

**"Bitcoins: {bitcoins}"**

**"Health: {health}"**

### Input / Constraints

You receive a **string** representing the dungeon's 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 bitcoins.  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 bitcoins.  You slayed snake.  You found 110 bitcoins.  You've made it!  Bitcoins: 120  Health: 65 |

# Problem 3 - Heart Delivery

Problem for exam preparation for the [Programming Fundamentals Course @SoftUni](https://softuni.bg/trainings/3447/programming-fundamentals-with-csharp-september-2021).

Submit your solutions in the SoftUni judge system at [https://judge.softuni.org/Contests/Practice/Index/2031#2](https://judge.softuni.org/Contests/Practice/Index/2031" \l "2).

*Valentine's day is coming, and Cupid has minimal time to spread some love across the neighborhood. Help him with his mission!*

You will receive a **string** with **even integers,** separated by a **"@"** - this is our neighborhood. After that, a series of **Jump** commands will follow until you receive **"Love!"**.Every house in the neighborhood needs a certain number of **hearts** delivered by Cupid so it can celebrate Valentine's day. The integers in the neighborhood indicate those needed hearts.

Cupid starts at the position of the **first** **house** (index 0) and must jump by a **given length.** The jump commands will be in this format: **"Jump {length}"**.

Every time he jumps from one house to another, the needed hearts for the visited house are **decreased by 2**:

* If the needed hearts for a certain house become **equal to 0**, print on the console **"Place {house\_index} has Valentine's day."**
* If **Cupid** jumps to a house where the needed hearts are **already** **0,** print on the console **"Place {house\_index} already had Valentine's day."**
* Keep in mind that **Cupid** can have a **larger jump length** than the **size of the neighborhood,** and if he does jump **outside** of it, he should **start** from the **first house** again (index 0)

*For example, we are given this neighborhood: 6@6@6. Cupid is at the start and jumps with a length of 2. He will end up at index 2 and decrease the needed hearts by 2: [6, 6, 4]. Next, he jumps again with a length of 2 and goes outside the neighborhood, so he goes back to the first house (index 0) and again decreases the needed hearts there: [4, 6, 4].*

### Input

* On the first line, you will receive a **string** with **even integers** separated by **"@"** –the neighborhood and the number of hearts for each house.
* On the next lines, until "**Love!**" is received, you will be getting jump commands in this format: "**Jump {length}**".

### Output

In the end, print **Cupid's** **last position** and whether his mission was successful or not:

* "**Cupid's last position was {last\_position\_index}.**"
* If **each house** has had Valentine's day, print:
  + "**Mission was successful.**"
* If **not,** print the **count** of all houses that **didn't** celebrate Valentine's Day:
  + **"Cupid has failed {houseCount} places."**

### Constraints

* The **neighborhood's** size will be in the range [1…20]
* Each **house** will need an **even number** of hearts in the range [2 … 10]
* Each **jump length** will be an integer in the range [1 … 20]

### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 10@10@10@2  Jump 1  Jump 2  Love! | Place 3 has Valentine's day.  Cupid's last position was 3.  Cupid has failed 3 places. | Jump 1 ->> [10, 8, 10, 2]  Jump 2 ->> [10, 8, 10, 0] so we print "Place 3 has Valentine's day."  The following command is "Love!" so we print Cupid's last position and the outcome of his mission. |
| 2@4@2  Jump 2  Jump 2  Jump 8  Jump 3  Jump 1  Love! | Place 2 has Valentine's day.  Place 0 has Valentine's day.  Place 0 already had Valentine's day.  Place 0 already had Valentine's day.  Cupid's last position was 1.  Cupid has failed 1 places. |  |