# CPP Coding Problem

## Subject: Weapon and Gems

## Contributor: 孫上晏, Yen-Chen Chiu

**Main testing concept:**

Code Comprehension, Graph, Interface

**Basics**

■ C++ BASICS
■ FLOW OF CONTROL
■ FUNCTION BASICS
■ PARAMETERS AND OVERLOADING
■ ARRAYS
■ STRUCTURES AND CLASSES
■ CONSTRUCTORS AND OTHER TOOLS
■ OPERATOR OVERLOADING, FRIENDS,AND REFERENCES
■ STRINGS
■ POINTERS AND DYNAMIC ARRAYS

**Functions**

□ SEPARATE COMPILATION AND NAMESPACES
□ STREAMS AND FILE I/O
■ RECURSION
■ INHERITANCE
■ POLYMORPHISM AND VIRTUAL FUNCTIONS
□ TEMPLATES
□ LINKED DATA STRUCTURES
■ EXCEPTION HANDLING
□ STANDARD TEMPLATE LIBRARY
□ PATTERNS AND UML

## Description:

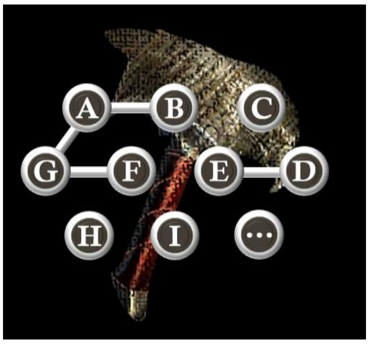
Please implement a Weapon&Gems system that can let a player arbitrarily place or remove gems on the weapon's sockets. Gems that are attached to a weapon can make the weapon more powerful. Some gems may provide skills for a weapon to cast, while the other gems affect those skills. When a skill is cast, the system should calculate the damage value according to the gems' connections.

In this system, there are 3 parts to implement. Your virtual teammate, Ian, had already implemented the first part and second part (green marked). Please implement the third part's functions (moving gems, damage calculation) described below.

There are 4 definitions in the following paragraph describing how the system works.

- **[Weapon Definition]**
  Player will only have 1 weapon. A weapon have the following attributes:
  - **Name**
  - **BaseDamage**
  - **ElementalDamage**

  A weapon has a total 26 sockets for placing gems (named from "A" to "Z").
  Sockets may be connected to other sockets. The connections will be defined in the initialization.
  Each socket can place one "gem" (defined below), and the gem in the socket can be removed or be replaced by another gem.



- **[Gem Definition]**
  There are 2 types of gems, both of the gems can be placed on the weapon's socket.
  1. Skill-gem:
     A skill-gem can cast an attack skill, which will deal damage. The damage calculation should follow the [Damage Definition].
     A skill-gem has the following attributes:
     - ○ **GemName** (skill name) (e.g., "Fireball")

- ○ **ElementalDamage** (e.g., 499)
- ○ **SkillDamage** (e.g., 13)

2. Support-gem:
   A support-gem cannot be cast, but it will affect all the connected skill-gems.
   A support-gem have the following attributes:
   - ○ **GemName** (e.g., "Combustion")
   - ○ **ElementalDamage** (e.g., 11)
   - ○ **BonusDamage** (e.g., 2)

- **[Damage Definition]**
  - The damage value of casting a skill-gem =

**(BaseDamage + Σ(ElementalDamage)) * SkillDamage * (1 + Σ(BonusDamage))**

  - Only skill-gems that are placed on the weapon can be cast and deal damage.
  - All the skill-gems connected with the casting skill-gem have to apply their **ElementalDamage** to the damage calculation.
  - All the support-gems connected with the casting skill-gem have to apply their **ElementalDamage** and **BonusDamage** to the damage calculation.

- **[User Control Definition]**
  The player will only do 2 things:
  1. Place the gem on the weapon's socket or remove it from the socket.
  2. Cast a skill-gem.

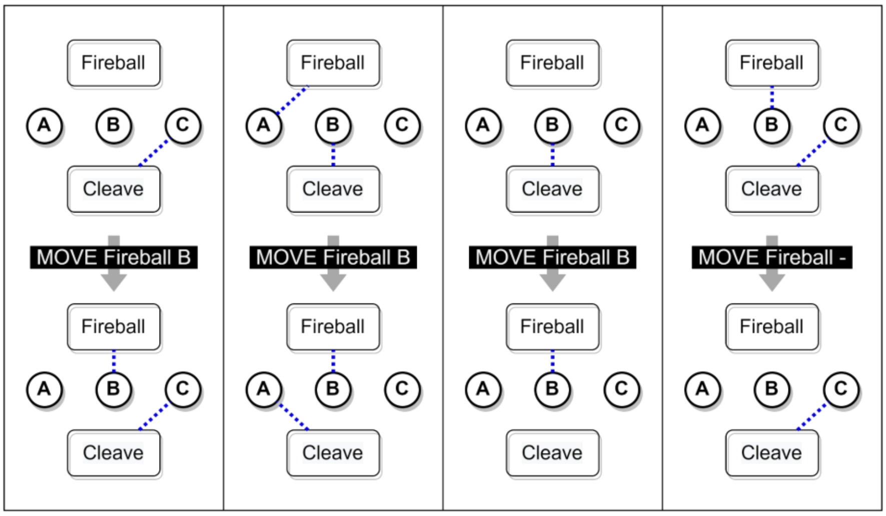
## Input / Output:

- [ First Part - Weapon Initialization ]
  1. The first line will input 2 ***integers*** to set the weapon's base damage and elemental damage.
     The format will be:
     **<BaseDamage> <ElementalDamage>**
  2. The second line will input an ***integer*** **<ConnectCount>** to decide how many following lines will be input to define the connectivity of sockets.
  3. There'll be N lines (N = <ConnectCount>) being input after the second line. Each line will input 2 ***characters*** (between A~Z), meaning there's a connection between these 2 sockets.
     The format will be:
     **<SocketName1> <SocketName2>**

- [ Second Part - Gems Initialization ]
  1. The first line will input an integer **<GemCount>** to decide how many following lines will be input to define the gems.
  2. There'll be M lines (M = <GemCount>) being input after the second line. Each line will input 4 parameters to initialize a gem.
     The format will be:
     **<GemType> <GemName> <GemAttribute1> <GemAttribute2>**
     - ◆ <**GemType**> and <**GemName**> are ***strings***
     - ◆ <**GemAttribute1**> and <**GemAttribute2**> are ***integers***
     - ◆ If the <**GemType**> is "**SkillGem**", then:
       - This gem is a skill-gem named **<GemName>**
       - **<GemAttribute1>** represents **"ElementalDamage"**
       - **<GemAttribute2>** represents **"SkillDamage"**
     - ◆ If the <**GemType**> is "**SupportGem**", then:
       - This gem is a support-gem named **<GemName>**
       - **<GemAttribute1>** represents **"ElementalDamage"**
       - **<GemAttribute2>** represents **"BonusDamage"**

3. The following line will remain empty for aesthetics.

- [ Third Part - Player Interaction ]
  After the second part, every input belongs to the third part. The system should be able to keep handling inputs until EOF is met.
  For each line, according to the first input ***string*** **<Mode>**, we can separate the interaction into 2 modes:
  1. <Mode> = "**MOVE**":
     The player is **moving a gem**. The **<GemName>** and **<SocketName>** will be input in the same line, which means moving the gem named **<GemName>** to the socket named **<SocketName>**.
     The format will be:
     **MOVE <GemName> <SocketName>**
     ◆ In the beginning, all the gems are not placed in any weapon's socket.
     ◆ If the **<SocketName>** is "-" and the gem is in a weapon's socket, the gem should be removed from the weapon's socket.
     ◆ If there's already a gem in the socket named **<SocketName>**, the system should **swap** the gems between their positions.
     ◆ The image below shows the example of how the MOVE instruction should be handled between different status of sockets.



  2. <Mode> = "**CAST**":
     The player is casting a skill-gem. The **<GemName>** will be input in the same line, which means casting the skill-gem named **<GemName>**.
     The format will be:
     **CAST <GemName>**
     ◆ When a skill-gem is cast, the system should output a line with the casting skill name followed by a space and the calculated damage value.
     For example: "**[InfernalBlow] 152**".
     ◆ If the casting skill-gem is not placed in any socket, the system should output a line with the casting skill name followed by a space and the message: "Gem is not placed in any socket".
     For example: "**[InfernalBlow] Gem is not placed in any socket**".

## Sample Input / Output :

| Sample Input | Sample Output |
|---|---|
| 17 3<br>4<br>A B<br>D E | [Fireball] 6747<br>[Cleave] Gem is not placed in any socket<br>[Fireball] 6747<br>[Fireball] 20670 |

```
F G
A G
4
SkillGem Fireball 499 13
SkillGem Cleave 7 100
SupportGem Combustion 11 2
SupportGem Multistrike 0 5

MOVE Fireball B
CAST Fireball
CAST Cleave
MOVE Combustion C
CAST Fireball
MOVE Combustion G
CAST Fireball
MOVE Multistrike F
MOVE Cleave D
CAST Fireball
CAST Cleave
MOVE Combustion -
CAST Fireball
MOVE Combustion F
MOVE Fireball D
CAST Fireball
CAST Cleave
MOVE Fireball F
CAST Fireball
CAST Cleave
MOVE Multistrike G
CAST Fireball
CAST Cleave
MOVE Combustion A
CAST Fireball
CAST Cleave
```

```
[Fireball] 55120
[Cleave] 2700
[Fireball] 40482
[Fireball] 6747
[Cleave] 11400
[Fireball] 6838
[Cleave] 52600
[Fireball] 41028
[Cleave] 315600
[Fireball] 55848
[Cleave] 429600
```

□ **Eazy,Only basic programming syntax and structure are required.**

■ **Medium,Multiple programming grammars and structures are required.**

□ **Hard,Need to use multiple program structures or complex data types.**

## Expected solving time:

45 minutes

## Other notes:

- The name of gems (including skill-gems and support-gems) will not be repeated in the Gems Initialization part.
- All the numerical values in this system will not exceed the range of *Integer(32-bit)*.
- In the third part, the input <GemName> in the "MOVE" case will always match one declared gem named <GemName>.
- In the third part, the input <GemName> in the "CAST" case will always match one declared skill-gem named <GemName>.
- There's no need to consider any influence between different elements (e.g., ice, fire, …) included in the "**ElementalDamage**" attribute.
- It's free to create or modify any code since any file will not be replaced when submitted to

the online judge.

- There’s a class diagram “ClassDiagram.png” in the downloaded file as a reference for the existing code.
- The status of blue marked “CAST Cleave”:
  - gem position:
    [Fireball] in D
    [Cleve] in B
    [Combustion] in F
    [Multistrike] in no socket
  - the damage of casting [Cleve] *(connected with [Combustion])*
    = (17 + (3+7+11)) * 100 * (1 + (2))
    = 11400
- The status of purple marked “CAST Cleave”:
  - gem position:
    [Fireball] in F
    [Cleve] in B
    [Combustion] in A
    [Multistrike] in G
  - the damage of casting [Cleve] *(connected with [Combustion], [Multistrike], [Fireball])*
    = (17 + (3+7+499+0+11)) * 100 * (1 + (5+2))
    = 429600
- This system is inspired by an online ARPG: Path of Exile.