CODEFORCES
Sponsored by TON

| k.ksu | Выйти

ГЛАВНАЯ ТОП КАТАЛОГ СОРЕВНОВАНИЯ ТРЕНИРОВКИ АРХИВ ГРУППЫ РЕЙТИНГ EDU API КАЛЕНДАРЬ ПОМОЩЬ

ЗАДАЧИ ОТОСЛАТЬ МОИ ПОСЫЛКИ ПОЛОЖЕНИЕ ЗАПУСК

A. Assignment 2

time limit per test: 1 second
memory limit per test: 256 megabytes
input: input.txt
output: output.txt

Overview: In this assignment, you will develop a C++ program that simulates a medieval fantasy story. Your program will create characters, weapons, and spells that interact within a dynamic story. The story progresses through a series of events such as attacks, spawns, and deaths, with each event written to an output file to narrate the unfolding adventure.

Entities These are the entities in this task:

- Narrator.
- Characters: *Character*, *Fighter*, *Wizard*, *Archer*.
- Physical items: *PhysicalItem*, *Weapon*, *Potion*, *SpellSheet* (Spell).
- Containers: *Container*, *Arsenal* (a container of weapons), *MedicalBag* (a container of potions), *SpellBook* (a container of SpellSheets).

Each event causes some text in the output file.
Specifications:

- One Narrator should be created, at the beginning.
- It's guaranteed that no character is going to be named Narrator.
- Names of items and characters are unique.
- Physical Item with owner property.

  1. Weapon: each weapon has a certain damage value.
  2. Potion: drink increases HP. Each potion has a certain healing value. A character can drink another character's potion.
  3. SpellSheet. Kills one of a given set of characters. A character can cast a spell as long as they have the associated SpellSheet in their SpellBook.

  - When cast, the spell causes its sheet to burn (the owner can't use it anymore). A potion can be drunk once. Unlike weapons, each potion and spell can be used only once, whereas weapons can be used indefinitely.

- Container

  1. Arsenal (a container of weapons). Each Arsenal has a different set of weapons.
  2. MedicalBag (a container of potions). Each MedicalBag has a different set of potions.
  3. SpellBook (a container of spells). Each SpellBook has a different set of spellSheets.

- Character has health points.

  1. Fighter:
     - Arsenal of 3 Weapons at max.
     - MedicalBag of 5 potions at max.
  2. Wizard: can cast spell and drink
     - SpellBook of 10 spells at max.
     - MedicalBag of 10 potions at max.
  3. Archer:
     - Arsenal of 2 weapons at max.
     - MedicalBag of 3 potions at max.
     - SpellBook of 2 spells at max.

IU SSAD Spring 2024
Закрытая
Участник

→ Соревнования группы

- SSAD_Assignment_4
- SSAD_Assignment_3
- SSAD_Assignment_2
- SSAD_Assignment_1

SSAD_Assignment_2
Закончено
Участник

→ Языки

Только перечисленные языки могут быть использованы для решения задач соревнования

SSAD_Assignment_2:

- GNU G++14 6.4.0
- GNU G++17 7.3.0
- GNU G++20 13.2 (64 bit, winlibs)

→ Виртуальное участие

Виртуальное соревнование – это способ прорешать прошедшее соревнование в режиме, максимально близком к участию во время его проведения. Поддерживается только ICPC режим для виртуальных соревнований. Если вы раньше видели эти задачи, виртуальное соревнование не для вас – решайте эти задачи в архиве. Если вы хотите просто дорешать задачи, виртуальное соревнование не для вас – решайте эти задачи в архиве. Запрещается использовать чужой код, читать разборы задач и общаться по содержанию соревнования с кем-либо.

Начать виртуальное участие

→ Последние посылки

Посылка	Время	Вердикт
255158027	06.04.2024 00:48	**Полное решение: 50 баллов**
255151282	05.04.2024 23:22	Частичное решение: 42 баллов
255149954	05.04.2024 23:08	Частичное решение: 42 баллов
255149781	05.04.2024 23:06	Частичное решение: 23 баллов

- When a character dies, all their items get destroyed.
- A dead character should be considered a non-existent character in the subsequent events after his/her death.

We strongly encourage you to use declaration and definition principles. The `main` function in your solution shouldn't have direct access (pointers) to items or containers. It should have access to characters only. Deleting a character destroys all his/her items.
Errors to be handled:

- Character doesn't exist (Attack, Cast, Drink, Create item, spell target is not a character, Dialogue)
- Character doesn't own an item (Attack, Cast, Drink).
- Negative value or zero for a potion healValue (Create item potion).
- Negative value or zero for a weapon's damageValue (Create item weapon).
- Target is not in the list of allowed targets in casting spells (Cast).
- Arsenal, MedicalBag, or SpellBook are full (Create item)
- Character can't carry or use a certain item, e.g., wizards can't have weapons (show $itemType $characterName, Create Item, Attack, Cast).

The input is guaranteed to be valid except for the possible errors mentioned above. In the output file in case of any of the previous errors, just add a line which contains "Error caught" and continue the execution of the program skipping the event that caused the error.
Requirements: You are required to meet the following criteria for the design specifics:

1. Design the classes Character, PhysicalItem, and Container such that the deletion of a Character results in the deletion of all his/her containers and items (10%).
2. Add Concept for objects derived from PhysicalItem (10%).
3. Design Container as a class template with specialization for objects derived from PhysicalItem with the help of the previous concept. The specialization improves the performance by leveraging items' unique names (10%).
4. The use of non-smart pointers will result in obtaining **zero** points for the design specifics component.

> You are given a UML diagram representing a recommendation for your solution. You are allowed to come up with your design. As long as it satisfies the design specifics mentioned above, no deduction will be applied. Here's the link to UML diagram. Use this site to view the UML.

Input

The input file contains $(n+1)$ lines where $1 \leq n \leq 2000$. The first line of the input contains a single integer denoting n. The number n represents the number of events in the story. Each of the next n lines contains an event. Below is the format of each input. The format followed is `[T]N`, where T is the type of the input value and N is a descriptive name of the value and what it represents. Everything inside the parentheses "()" is an additional event description. Please refer to the examples in case of unclarity or feel free to ask questions on the assignment's forum on Moodle. Each name, whether it be a character name or an item name, consists of lowercase latin letters and its length is at max 10.

1. "Create character `$[string]type $[string]name $[int]initHP`" (`type` can be fighter, wizard, or archer. $1 \leq \texttt{initHP} \leq 200$).
2. "Create item weapon `$[string]ownerName $[string]weaponName $[int]damageValue`" ($1 \leq \texttt{damageValue} \leq 50$).
3. "Create item potion `$[string]ownerName $[string]potionName $[int]healValue`" ($1 \leq \texttt{healValue} \leq 50$).
4. "Create item spell `$[string]ownerName $[string]spellName $[int]m $[string_1, string_2, ..., string_m]characterNames`" (set of $0 \leq m \leq 50$ unique characters names the spell can be applied on).
5. "Attack `$[string]attackerName $[string]targetName $[string]weaponName`" (attacker and target can be the same).
6. "Cast `$[string]casterName $[string]targetName $[string]spellName`" (caster and target can be the same).
7. "Drink `$[string]supplierName $[string]drinkerName $[string]potionName`" (supplier and drinker can be the same).
8. "Dialogue `$[string]speaker` (a character name or "Narrator") `$[int]sp_len $[string_1, string_2, ..., string_sp_len]speech`" (the speech itself where its length is `sp_len`, $1 \leq \texttt{sp\_len} \leq 10$).
9. "Show characters" (show alive characters in alphabetical order).

255149722	05.04.2024 23:05	Частичное решение: 42 баллов
255149242	05.04.2024 23:00	Частичное решение: 42 баллов
255148506	05.04.2024 22:51	Частичное решение: 19 баллов
255147454	05.04.2024 22:40	Частичное решение: 15 баллов
255146892	05.04.2024 22:34	Неправильный ответ на тесте 1
255146853	05.04.2024 22:33	Ошибка компиляции

→ Набранные баллы

	Баллы
A	50
Всего	**50**

10. "Show weapons `$[string]characterName`" (show weapons owned by `characterName` in alphabetical order).
11. "Show potions `$[string]characterName`" (show potions owned by `characterName` in alphabetical order).
12. "Show spells `$[string]characterName`" (show spells owned by `characterName` in alphabetical order).
13. The death of characters should be handled automatically.

- Please note that the constraints mentioned in this section are for generated large tests, and they don't apply in the samples. The constraints should be used for deciding algorithms and the data types in your solution, e.g., choose `int` vs `long`. Sample 2 has a spell name (AvadaKedavra) longer than 10. Sample 3 potions have too big heal values (>50) compared to the constraint given. Your solution should pass these sample tests (as they have points). It is guaranteed the constraints hold for all other tests.

Output

The ordered list of outputs matches the events in the input. Each event output is on its own line. Note that the number of lines is undetermined before executing the code due to the death event.

1. Expected output: "A new `$[string]type` came to town, `$[string]name`."
2. Expected output: "`$[string]ownerName` just obtained a new weapon called `$[string]weaponName`."
3. Expected output: "`$[string]ownerName` just obtained a new potion called `$[string]potionName`."
4. Expected output: "`$[string]ownerName` just obtained a new spell called `$[string]spellName`."
5. Expected output: "`$attackerName` attacks `$targetName` with their `$weaponName`!"
6. Expected output: "`$casterName` casts `$spellName` on `$targetName`!"
7. Expected output: "`$drinkerName` drinks `$potionName` from `$supplierName`."
8. Expected output: "`$speaker`: `$speech`"
9. Split characters by a single space. Each character should be of the format `$[string]name`:`$[string]type`:`$[int]currentHealthPoints`.
10. Split weapons by a single space. Each weapon in the output should be of the form `$[string]weaponName`:`$[int]weaponDamage`.
11. Split potions by a single space. Each potion in the output should be of the form `$[string]potionName`:`$[int]potionHealValue`.
12. Split spells by a single space. Each spell in the output should be of the form `$[string]spellName`:`$[int]numAllowedTargets`.
13. Death expected output: "`$[string]charName` has died..."

Examples

input Скопировать

```
13
Dialogue Narrator 4 Once upon a time
Create character fighter Arthur 200
Create item weapon Arthur Excalibur 50
Attack Arthur John Excalibur
Create character archer Robin 150
Create character wizard Harry 100
Attack Arthur Harry Excalibur
Show characters
Attack Arthur Harry Excalibur
Show characters
Show weapons Arthur
Show potions Arthur
Show spells Arthur
```

output Скопировать

```
Narrator: Once upon a time
A new fighter came to town, Arthur.
Arthur just obtained a new weapon called Excalibur.
Error caught
A new archer came to town, Robin.
A new wizard came to town, Harry.
Arthur attacks Harry with their Excalibur!
Arthur:fighter:200 Harry:wizard:50 Robin:archer:150
Arthur attacks Harry with their Excalibur!
Harry has died...
Arthur:fighter:200 Robin:archer:150
Excalibur:50
```

```
Error caught
```

input Скопировать

```
5
Dialogue Narrator 4 Once upon a time
Create character fighter Arthur 200
Create character wizard Harry 50
Create item spell Harry AvadaKedavra 2 Harry Arthur
Cast Harry Arthur AvadaKedavra
```

output Скопировать

```
Narrator: Once upon a time
A new fighter came to town, Arthur.
A new wizard came to town, Harry.
Harry just obtained a new spell called AvadaKedavra.
Harry casts AvadaKedavra on Arthur!
Arthur has died...
```

input Скопировать

```
12
Dialogue Narrator 4 Once upon a time
Create character fighter Arthur 200
Create item potion Arthur elixir1 50
Create item potion Arthur elixir2 100
Create item potion Arthur elixir3 150
Create item potion Arthur elixir4 250
Create item potion Arthur elixir5 350
Create item potion Arthur elixir6 450
Dialogue Arthur 4 Time for some elixir
Drink Arthur Arthur elixir1
Show characters
Show potions Arthur
```

output Скопировать

```
Narrator: Once upon a time
A new fighter came to town, Arthur.
Arthur just obtained a new potion called elixir1.
Arthur just obtained a new potion called elixir2.
Arthur just obtained a new potion called elixir3.
Arthur just obtained a new potion called elixir4.
Arthur just obtained a new potion called elixir5.
Error caught
Arthur: Time for some elixir
Arthur drinks elixir1 from Arthur.
Arthur:fighter:250
elixir2:100 elixir3:150 elixir4:250 elixir5:350
```

Note

Some useful links:

1. https://en.cppreference.com/w/cpp/io
2. https://en.cppreference.com/w/cpp/language/derived_class
3. https://en.cppreference.com/w/cpp/memory/enable_shared_from_this
4. https://en.cppreference.com/w/cpp/language/friend
5. https://en.cppreference.com/w/cpp/memory/shared_ptr/pointer_cast
6. https://en.cppreference.com/w/cpp/utility/variant

При поддержке

УНИВЕРСИТЕТ ИТМО