ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

Kỹ thuật Lập trình - CO1027

Bài tập lớn 2

THANH GƯƠM TRONG ĐÁ

Phiên bản 1.0

TP. HỒ CHÍ MINH, THÁNG 03/2023

ĐẶC TẢ BÀI TẬP LỚN

Phiên bản 1.0

1 Chuẩn đầu ra

Sau khi hoàn thành bài tập lớn này, sinh viên ôn lại và sử dụng thành thục:

- Con trỏ.
- Lập trình hướng đối tượng.
- Danh sách liên kết đơn.

2 Dẫn nhập

Ultimecia, nữ phù thuỷ hùng mạnh nhất mọi thời đại, do đau khổ trong tình cảm, đã quyết định tiêu diệt toàn thể nhân loại bằng phép nén thời gian; qua đó quá khứ, hiện tại và tương lai sẽ được hợp nhất. Không có quá khứ, hiện tại và tương lai, sẽ không còn hoài niệm, khát vọng và phát triển, nhân loại sẽ không còn tồn tại.

Để thi hành phép nén thời gian, Ultimecia trồng một bông hồng ma thuật. Các cánh hoa hồng sẽ rơi dần theo thời gian. Khi cánh hoa hồng cuối cùng rơi xuống, phép nén thời gian sẽ hoàn thành. Để đảm bảo an toàn cho bông hồng, Ultimecia giấu bông hồng sau mê cung bóng tối với nhiều quái vật để ngăn cản bất kì Hiệp sĩ nào đến gần. Nếu có hiệp sĩ nào vượt qua được mê cung bóng tối và đánh bại được Ultimecia ở cuối mê cung thì nàng sẽ hủy bỏ quyết định và nhân loại sẽ được giải cứu khỏi thảm họa diệt vong.

Muốn đánh bại Ultimecia, cần phải có được thanh kiếm Excalibur của vua Arthur. Vua Arthur sau khi đánh bại Bowser và cứu được công chúa Guinevere , đã cắm lại thanh kiếm Excalibur vào đá; từ giã ngai vàng và cuộc đời hiệp sĩ, cùng công chúa Guinevere ẩn thân nơi miền hoang dã. Muốn rút lại được thanh kiếm Excalibur ra khỏi đá, cần phải có được ba báu vật: chiếc khiên của Paladin, ngọn giáo của Lancelot và một sợi tóc của Guinevere. Các báu vật này được giấu ở nhiều nơi khác nhau trong mê cung bóng tối.

Ngay cả khi được rút khỏi đá, gươm Excalibur chỉ có thể được sử dụng bởi một hiệp sĩ Bàn Tròn. Hơn nữa, một hiệp sĩ đơn độc không thể đánh bại được vô số quái vật cùng phép thuật mạnh mẽ của Ultimecia. Vì vậy, một Đội quân các Hiệp sĩ tài giỏi đã tập hợp lại để thực hiện nhiệm vụ mới đầy thử thách và vinh quang: tìm lại thanh gươm trong đá, chiến

thắng Ultimecia và giải cứu nhân loại. Liệu các hiệp sĩ có rút được thanh gươm trong đá, liệu Ultimecia có từ bỏ ý định cực đoan trong đau khổ của mình. Tất cả sẽ được giải đáp trong bài tập lớn này.

3 Dữ liệu nhập

Dữ liệu nhập của chương trình được chứa trong 2 file nhập dữ liệu cho Đội quân Hiệp sĩ và các sự kiện trong hành trình đánh bại Ultimecia.

3.1 Đội quân Hiệp sĩ

Tên file dữ liệu cho Đội quân Hiệp sĩ sẽ được lưu trong biến **file_armyknights**. File này có định dạng như sau:

- Dòng 1 chứa 1 số nguyên dương n, $n <= 1000$, biểu diễn cho số lượng hiệp sĩ Bàn Tròn trong Đội quân Hiệp sĩ chiến đấu với Ultimecia.
- n dòng tiếp theo, mỗi dòng là một chuỗi mô tả thông tin cho một Hiệp sĩ Bàn Tròn. Chuỗi có định dạng như sau:

  ```
  HP␣level␣phoenixdownI␣gil␣antidote
  ```

  Trong đó:

  - ***HP***: Chỉ số sức khỏe của hiệp sĩ, là một số nguyên có giá trị từ 1 đến 999. Giá trị này cũng là giá trị sức khỏe tối đa ***MaxHP*** của hiệp sĩ.
  - ***level***: Đẳng cấp của hiệp sĩ, là một số nguyên có giá trị từ 1 đến 10.
  - ***phoenixdownI***: Số giọt nước mắt phượng hoàng PhoenixDown loại 1 mà hiệp sĩ mang theo, là 1 số nguyên có giá trị từ 0 đến 5.
  - ***gil***: Số tiền mà hiệp sĩ mang theo, là một số nguyên có giá trị từ 1 đến 999.
  - ***antidote***: Số thuốc giải mà hiệp sĩ mang theo, là một số nguyên có giá trị từ 0 đến 5.

3.2 Sự kiện

Tên file dữ liệu nhập cho các sự kiện sẽ được lưu trong biến **file_events**. File này có định dạng như sau:

- Dòng 1 chứa 1 số nguyên dương e, $e <= 1000$, biểu diễn cho số lượng các sự kiện mà đội quân hiệp sĩ sẽ gặp.
- Dòng tiếp theo chứa e sự kiện, mỗi sự kiện được đánh chỉ số bắt đầu từ 1. Mỗi sự kiện sẽ được mô tả bằng một giá trị số, gọi là mã sự kiện. Ý nghĩa tương ứng của từng sự kiện được mô tả trong Mục 4. Một sự kiện có thể xảy ra nhiều lần. Ultimecia sẽ đứng trước và bảo vệ bông hồng khỏi đội quân Hiệp sĩ. Do đó, sự kiện mã 99 gặp Ultimecia chỉ xuất hiện 1 lần và là sự kiện cuối cùng. Các sự kiện 95, 96, 97, 98 sẽ đảm bảo xảy ra ít nhất một lần.

chưa hiện thực

4 Mô tả

Hành trình chiến đấu với Ultimecia đi qua nhiều sự kiện. Ý nghĩa tương ứng của từng sự kiện được mô tả trong Bảng 1. Tùy theo các sự kiện diễn ra trên đường đi mà thông tin của Đội quân Hiệp sĩ sẽ có sự thay đổi. Cách thức chiến đấu của Đội quân là Hiệp sĩ ở cuối sẽ chiến đấu với đối thủ, tham khảo thêm Mục 5.3. Nếu gặp vật phẩm, hiệp sĩ ở cuối sẽ cố gắng đặt vật phẩm đó vào túi đồ. Nếu không đặt vật phẩm vào túi đồ được, Hiệp sĩ sẽ đưa vật phẩm cho Hiệp sĩ đứng trước và Hiệp sĩ đứng trước này tiếp tục cố gắng đưa vào túi đồ. Quá trình này lặp lại cho đến Hiệp sĩ đầu tiên trong Đội quân. Nếu Hiệp sĩ đầu tiên cũng không thể đưa vào túi đồ thì vật phẩm đó sẽ bị bỏ đi. Bên cạnh đó, nếu Hiệp sĩ cuối nhận được gil quá 999 thì phần gil dư cũng sẽ được truyền lại cho Hiệp sĩ đứng trước như cách truyền lại vật phẩm.

single linked list ?

Thông tin của các sự kiện như sau:

1. **Nếu gặp sự kiện có mã từ 1 đến 5**, hiệp sĩ phải giao tranh với đối thủ tương ứng. Nếu level của hiệp sĩ cao hơn hoặc bằng levelO của đối thủ, hiệp sĩ chiến thắng. Mỗi khi chiến thắng đối thủ, hiệp sĩ sẽ nhận được một số tiền tương ứng, được mô tả trong 2, tuy nhiên số gil của hiệp sĩ không bao giờ vượt quá 999.
   Nếu level của hiệp sĩ nhỏ hơn levelO của đối thủ, HP của hiệp sĩ sẽ được tính lại như sau:

$$HP = HP - baseDamage * (levelO - level) \quad (1)$$

   Trong đó, baseDamage sẽ tuỳ thuộc vào đối thủ, được mô tả ở Bảng 3, $levelO$ của Đối thủ được tính như sau:

$$levelO = (i + event_{id})\%10 + 1$$

   Trong đó, i là thứ tự của sự kiện bắt đầu tính từ 0, $event_{id}$ là mã của sự kiện.

tham khảo BTL 1

xanh lá: khác BTL1

xanh dương: check coi khác ko

Bảng 1: Các sự kiện trên hành trình đến Ultimecia

Mã sự kiện	**Ý nghĩa**
1	Gặp gấu MadBear
2	Gặp cướp Bandit
3	Gặp tướng cướp LordLupin
4	Gặp yêu tinh Elf
5	Gặp quỷ khổng lồ Troll
6	Gặp ma Tornbery
7	Gặp Nữ hoàng cờ bạn Queen of Cards
8	Gặp Lái buôn vui vui tính Nina de Rings
9	Gặp Vườn Sầu Riêng
10	Gặp quái vật Omega Weapon
11	Gặp Thần Chết Hades
112	Nhặt được giọt nước mắt phượng hoàng PhoenixDown loại II
113	Nhặt được giọt nước mắt phượng hoàng PhoenixDown loại III
114	Nhặt được giọt nước mắt phượng hoàng PhoenixDown loại IV
95	Nhặt được chiếc khiên của Paladin
96	Nhặt được ngọn giáo của Lancelot
97	Nhặt được sợi tóc của Guinevere
98	Gặp gươm Excalibur
99	Gặp Ultimecia

cái này dùng nhiều => viết hàm

Nếu HP nhỏ hơn hoặc bằng 0 sau khi tính bằng Công thức 1 thì chương trình cần thực hiện các bước sau:

single linked list ?

- Bước 1: Hiệp sĩ tìm PhoenixDown trong túi đồ theo thứ tự từ vị trí đầu đến cuối, với vị trí đầu chứa vật phẩm gần nhất mới nhận được. Nếu tìm thấy thì Hiệp sĩ sẽ sử dụng PhoenixDown để tăng HP. Nếu tìm hết túi đồ vẫn mà HP vẫn nhỏ hơn hoặc bằng 0 thì tiếp tục Bước 2.
- Bước 2: Nếu số gil của Hiệp sĩ đang lớn hơn hoặc bằng 100, Hiệp sĩ sẽ gọi Phượng hoàng đến để hồi sinh mình. Khi gọi phượng hoàng, số tiền của hiệp sĩ sẽ bị trừ đi 100 gil. Khi hồi sinh, HP của hiệp sĩ sẽ bằng 1/2 MaxHP (chỉ tính phần nguyên). Nếu hiệp sĩ không thể được hồi sinh do số gil nhỏ hơn 100 thì hiệp sĩ sẽ hy sinh, hiệp sĩ tiếp theo trong đội quân sẽ lên và chiến đối tiếp với đối thủ.

lấy phần nguyên bỏ thập phân ?

Bảng 2: Số gil thu được khi chiến thắng đối thủ

Đối thủ	gil
MadBear	100
Bandit	150
LordLupin	450
Elf	750
Troll	800

Bảng 3: Chỉ số baseDamage của các đối thủ

Đối thủ	baseDamage
MadBear	10
Bandit	15
LordLupin	45
Elf	75
Troll	95

2. **Nếu gặp ma Tornbery (mã sự kiện là 6)**, hiệp sĩ sẽ giao tranh với Tornbery. Cách giao tranh tương tự như mục đầu. Nếu hiệp sĩ thắng, level của hiệp sĩ sẽ tăng lên 1 đơn vị nhưng không được tăng quá 10. Nếu thua, hiệp sĩ sẽ bị trúng độc. Ngay khi bị trúng độc, nếu hiệp sĩ có thuốc antidote (antidote >= 1), hiệp sĩ sẽ tự động dùng thuốc này và trở lại bình thường, chỉ số antidote của hiệp sĩ bị giảm đi 1. Nếu không có thuốc antidote, hiệp sĩ sẽ bị độc ảnh hưởng như sau:
   - Hiệp sĩ làm rơi từ túi đồ 3 vật phẩm gần nhất nhặt được. Nếu túi đồ đang có nhỏ hơn hoặc bằng 3 vật phẩm thì túi đồ sẽ trở thành rỗng.

VẬY MẤY ITEM BAN ĐẦU THÌ KO RƠI ????????

- HP bị giảm đi 10, nếu HP trở thành nhỏ hơn hoặc bằng 0 thì thực hiện như mô tả mục đầu.

3. **Nếu gặp nữ hoàng cờ bạc Queen of Cards**, hiệp sĩ sẽ giao tranh với nữ hoàng. Cách giao tranh tương tự như mô tả trong mục đầu. Nếu hiệp sĩ thắng, gil của hiệp sĩ sẽ được tăng gấp đôi. Nếu số gil lớn hơn 999 thì chuyển phần gil dư ra cho Hiệp sĩ phía trước, Hiệp sĩ phía trước sẽ cố gắng nhận nhiều gil nhất có thể, nếu vẫn còn dư lại tiếp tục truyền về phía trước, quá trình này lặp lại đến hiệp sĩ đầu tiên, nếu hiệp sĩ đầu tiên không thể giữ được thêm thì phần gil dư sẽ bị bỏ đi. Nếu thua, hiệp sĩ sẽ bị giảm một nửa số gil (chỉ tính phần nguyên). lấy phần nguyên bỏ thập phân ?
4. **Nếu gặp lái buôn vui tính Nina de Rings**, hiệp sĩ thực hiện việc các công việc mua bán theo thứ tự được mô tả như sau: lấy phần nguyên bỏ thập phân ?
   - Hiệp sĩ sẽ tiếp tục hành trình và không mua bán gì cả nếu có ít hơn 50 gil.
   - Nếu HP của hiệp sĩ đang nhỏ hơn 1/3 MaxHP (chỉ tính phần nguyên), hiệp sĩ sẽ đưa Nina 50 gil sẽ giúp hiệp sĩ tăng HP thêm 1 lượng bằng 1/5 MaxHP.
5. **Nếu Đội quân Hiệp sĩ lạc vào Vườn Sầu Riêng**, HP của hiệp sĩ sẽ phục hồi về MaxHP. lấy phần nguyên bỏ thập phân ?
6. **Hiệp sĩ có thể gặp quái vật Omega Weapon**, một quái vật thời tiền sử tồn tại từ khi vũ trụ mới thành hình. Nếu gặp Omega Weapon, hiệp sĩ sẽ bại trận ở bất kỳ level nào. HP của hiệp sĩ sẽ giảm về 0 và phải thực hiện các bước như mô tả ở sự kiện đầu. Chỉ có hiệp sĩ ở level 10 và HP đang là MaxHP hoặc Hiệp sĩ Rồng ở level bất kỳ mới đánh thắng Omega Weapon. Trong trường hợp đánh thắng, level của Hiệp sĩ sẽ được tăng thành 10, gil của Hiệp sĩ được tăng thành 999. Sau khi bị đánh bại, Omega Weapon sẽ không bao giờ xuất hiện lại, nếu hiệp sĩ có gặp tiếp sự kiện có Omega Weapon thì hiệp sĩ sẽ bỏ qua sự kiện này và đi tiếp. nguy hiểm
7. **Nếu Đội quân Hiệp sĩ gặp Thần Chết Hades**, hiệp sĩ sẽ bại trận ở bất kì level nào dưới 10. Chỉ có hiệp sĩ ở level 10 hoặc Hiệp sĩ Paladin từ level 8 trở lên mới đánh thắng Hades. Nếu hiệp sĩ bại trận, HP sẽ giảm về 0 và phải thực hiện các bước như mô tả ở sự kiện đầu. Nếu đánh thắng Hades, Hades sẽ rèn cho hiệp sĩ chiếc khiên của Paladin. Nếu Đội quân Hiệp sĩ đã có chiếc khiên này thì sẽ không nhận thêm Khiên của Paladin. Mặt khác, nếu hiệp sĩ đã nhận Khiên từ Hades thì hiệp sĩ sẽ bỏ qua khiên của Paladin khi gặp sự kiện 95. Sau khi bị đánh bại, Hades sẽ không xuất hiện lại, nếu hiệp sĩ có gặp tiếp sự kiện có Hades thì hiệp sĩ sẽ bỏ qua sự kiện này và đi tiếp. nguy hiểm
   mình để nguyên luôn tại true rồi gặp nữa cũng true
   ko cần làm gì hết do cũng là gán hasPaladinShield =true
8. **Nếu Đội quân Hiệp sĩ gặp Ultimecia**, trận giao tranh sẽ diễn ra như sau. Nếu Đội quân Hiệp sĩ có gươm Excalibur, Hiệp sĩ sẽ đánh thắng Ultimecia.
   Nếu không có Gươm Excalibur nhưng có đủ 3 món bảo vật, Ultimecia sẽ đồng ý giao đấu

*tức là 3 loại hiệp sĩ đó sẽ lên lần lượt, mỗi hiệp sĩ đó sẽ làm HP Ultimecia giảm, nếu HP Ultimecia chưa = 0 thì Hiệp sĩ sẽ bị die.
*tới khi nào HP Ultimecia =0 mà vẫn còn hiệp sĩ thuộc 3 loại đó thì thắng ??? ngược lại thua ???

với các Hiệp sĩ Rồng, Paladin và Lancelot trong Đội quân. Mỗi Hiệp sĩ thuộc 1 trong 3 loại này trong đội quân sẽ lần lượt đấu với Ultimecia theo thứ tự từ Hiệp sĩ đứng cuối đến Hiệp sĩ đứng đầu. Cách thức giao đấu là mỗi Hiệp sĩ gây lên một lượng sát thương nhất định lên HP của Ultmecia. Nếu HP của Ultimecia bằng 0 trước khi Đội quân Hiệp sĩ không còn Hiệp sĩ nào có thể chiến đấu với Ultimecia thì Đội quân sẽ thắng, ngược lại đội quân thua. HP của Ultimecia ban đầu là **5000**. Mỗi loại Hiệp sĩ gây lượng sát thương như sau:

$$damage = HP * level * knightBaseDamage$$

Trong đó, damage chỉ lấy phần nguyên nếu kết quả là một số thực, knightBaseDamage được tính cho như Bảng 4. Sau khi gây sát thương cho Ultimecia, nếu chưa đánh bại Ultimecia thì Hiệp sĩ sẽ bị Ultimecia dùng phép thuật tấn công và hy sinh. Khi đó, hiệp sĩ phù hợp ở phía sau sẽ tiếp tục lên thay thế và tấn công Ultimecia.

Nếu Đội quân không có đủ 3 món bảo vật, thì đội quân sẽ thua Ultimecia. Khi đội quân thua, tất cả các Hiệp sĩ trong Đội quân sẽ hy sinh làm cho số lượng hiệp sĩ trong Đội quân bằng 0.

dùng số lượng hiệp sĩ hiện tại knights_count để return thắng thua

Bảng 4: Chỉ số knightBaseDamage của các loại hiệp sĩ

Hiệp sĩ	knightBaseDamage
Lancelot	0.05
Paladin	0.06
Hiệp sĩ Rồng	0.075

9. **Nếu Đội quân Hiệp sĩ gặp nước mắt phượng hoàng (mã sự kiện là 112, 113, 114**, các giọt nước mắt phượng hoàng này là từ các Hiệp sĩ trước đã hi sinh khi chiến đấu với Ultimecia. Vì đã ở lâu trong mê cung bóng tối, tác dụng của nước mắt phượng hoàng không còn như trước mà bị suy biến thành 3 loại khác nhau: II, III, IV. Sau đây là tác dụng của từng loại nước mắt phượng hoàng, bao gồm cả loại I là loại mà Hiệp sĩ mang theo trước khi bắt đầu hành trình:

gặp nào lấy đó

là loại I đem lúc đầu, lúc nhặt thì chỉ có II,III,IV ?

- Loại I: chỉ có thể dùng khi $HP <= 0$, phục hồi HP thành $MaxHP$.
- Loại II: có thể dùng khi $HP < MaxHP/4$ (chỉ tính phần nguyên), phục hồi HP thành $MaxHP$.
- Loại III: có thể dùng khi $HP < MaxHP/3$ (chỉ tính phần nguyên), nếu $HP <= 0$ thì phục hồi HP thành $MaxHP/3$ (chỉ lấy phần nguyên), ngược lại tăng thêm 1 lượng bằng $MaxHP/4$ (chỉ lấy phần nguyên).

CHƯA CODE

CHƯA CODE

- Loại IV: có thể dùng khi ***HP*** < ***MaxHP***/2 (chỉ tính phần nguyên), nếu ***HP*** <= 0 thì phục hồi ***HP*** thành ***MaxHP***/2 (chỉ lấy phần nguyên), ngược lại tăng thêm 1 lượng bằng ***MaxHP***/5 (chỉ lấy phần nguyên).

Trong quá trình đi qua các sự kiện, sau khi giao tranh và ***HP*** bị giảm, hiệp sĩ sẽ lần lượt tìm kiếm trong túi đồ của mình nước mắt phượng hoàng đầu tiên có thể sử dụng được và sử dụng nó ngay lập tức. Cách thức tìm kiếm là đi từ vật phẩm đầu tiên (tương ứng với đầu danh sách) đến vật phẩm cuối cùng (tương ứng với cuối danh sách), xem thêm cách hiện thực túi đồ bằng danh sách liên kết đơn ở Mục 5.6.

10. **Nếu Đội quân Hiệp sĩ gặp các bảo vật (mã sự kiện là 95, 96, 97)**, Đội quân sẽ nhặt các báu vật này.
11. **Khi Đội quân Hiệp sĩ gặp gươm Excalibur (mã sự kiện là 98)**, nếu chưa có đủ ba báu vật, hiệp sĩ sẽ không rút được thanh gươm này và sẽ tiếp tục hành trình đến sự kiện kế. Nếu đã có đủ ba báu vật, hiệp sĩ sẽ rút thanh gươm Excalibur và tiếp tục cuộc hành trình để tìm Ultimecia.
12. **Nếu Hiệp sĩ giao đấu là Lancelot**, hiệp sĩ sẽ đánh bại các đối thủ có mã từ 1 đến 5.
13. **Nếu Hiệp sĩ giao đấu là Hiệp sĩ Rồng**, do có Dòng máu Rồng nên Hiệp sĩ không bị trúng độc khi thua ma Tornbery.
14. **Nếu Hiệp sĩ giao đấu là Paladin**, hiệp sĩ sẽ đánh bại các đối thủ có mã từ 1 đến 5. Nếu thua Queen of Cards, Paladin không bị giảm số gil. Nina cũng sẽ đồng ý mua bán kể cả khi Paladin có ít hơn 50 gil và sẽ không thu 50 gil nếu giúp Paladin tăng HP.

gắn 3 mục này vô mấy cái liên quan để code 1 lần mấy cái đó

5 Các class trong chương trình

Bài tập lớn này sử dụng Lập trình Hướng đối tượng để mô tả Hành trình chiến đấu với Ultimecia. Các đối tượng trong Lập trình Hướng đối tượng được biểu diễn thông qua class và được mô tả như bên dưới.

5.1 Danh sách sự kiện

Class **Events** biểu diễn các sự kiện của hành trình chiến đấu với Ultimecia. Class này có các method (phương thức) như sau:

- Constructor có 1 tham số kiểu string, chứa tên file mô tả thông tin cho các sự kiện, xem lại Mục 3.2 về định dạng của file. Constructor này cấp phát một mảng các số nguyên để lưu trữ mã sự kiện. Khai báo của phương thức khởi tạo:

heap ?

```
Events(const string & file_events);
```

- Method **count** trả về số lượng sự kiện. Khai báo của phương thức:

```
int count() const;
```

- Method **get** trả về mã sự kiện ở vị trí i (vị trí tính từ 0).

```
int get(int i) const;
```

- Phương thức hủy (Destructor) cần thu hồi các vùng nhớ được cấp phát động trong Heap.

5.2 Hiệp sĩ

Trong Đội quân Hiệp sĩ đi chiến đấu với Ultimecia, có 4 loại Hiệp sĩ:

1. Paladin: Hiệp sĩ có HP ban đầu là một số nguyên tố.
2. Lancelot: Hiệp sĩ có HP ban đầu là 888.
3. Hiệp sĩ Rồng: Hiệp sĩ có HP ban đầu có đúng 3 chữ số và 3 chữ số này tạo thành bộ 3 số Pythagoras. Một bộ ba số Pythagoras gồm 3 số nguyên dương a, b, c sao cho $a^2 + b^2 = c^2$
4. Hiệp sĩ Bàn Tròn thông thường: Không là một trong các loại hiệp sĩ trên.

thằng này ko cần đánh dấu ??

Yêu cầu: Sinh viên hãy hiện thực các class sau để biểu diễn thông tin cho các Hiệp sĩ.

1. class **BaseKnight**: class (lớp) biểu diễn thông tin cho 1 hiệp sĩ. Class này có các thuộc tính với access modifier (phạm vi truy cập) là protected:
   - id: biểu diễn định danh cho Hiệp sĩ, kiểu int.
   - hp: chỉ số ***HP*** của Hiệp sĩ, kiểu int.
   - maxhp: chỉ số ***MaxHP*** của Hiệp sĩ, kiểu int.
   - level: ***level*** của Hiệp sĩ, kiểu int.
   - gil: số tiền đang có của Hiệp sĩ, kiểu int.
   - bag: biểu diễn túi đồ của Hiệp sĩ, kiểu BaseBag, sẽ được mô tả sau ở Mục 5.6.
   - knightType: biểu diễn loại Hiệp sĩ, kiểu enum KnightType gồm các giá trị: PALADIN, LANCELOT, DRAGON, NORMAL tương ứng với 4 kiểu Hiệp sĩ: Paladin, Lancelot, Hiệp sĩ Rồng và Hiệp sĩ Thông thường.

   class **BaseKnight** có 1 static method **create** nhận vào các chỉ số của Hiệp sĩ và tạo ra đối tượng Hiệp sĩ phù hợp. Khai báo của method này như sau:

```
static BaseKnight * create(int id, int maxhp, int level, int gil, int
    antidote, int phoenixdownI);
```

tức là dùng hàm này để tạo 1 hiệp sĩ và phân loại hiệp sĩ đó luôn ??? (do static thì có thể gọi thẳng trực tiếp từ lớp (ở đây là BaseKnight))

2. class **PaladinKnight**, class **LancelotKnight**, class **DragonKnight**, class **NormalKnight** lần lượt biểu diễn Hiệp sĩ kiểu Paladin, Lancelot, Hiệp sĩ Rồng và Hiệp sĩ Bàn Tròn thông thường. 4 class này đều kế thừa từ class BaseKnight và cần định nghĩa lại method **fight** cho phù hợp.

5.3 Đội quân Hiệp sĩ

Class **ArmyKnights** biểu diễn thông tin cho Đội quân gồm nhiều Hiệp sĩ. Các Hiệp sĩ này được lưu trữ bằng một mảng được cấp phát động. Class này có phương thức (method):

- Phương thức khởi tạo (Constructor) có 1 tham số kiểu string, chứa tên file mô tả thông tin cho Đội quân, xem thêm Mục 3.1 để xem dữ liệu nhập của file này. Constructor này khởi một mảng các Hiệp sĩ từ dữ liệu của file đọc vào. Id của Hiệp sĩ đầu tiên đọc vào là 1, id của mỗi Hiệp sĩ sau bằng Hiệp sĩ trước cộng thêm 1. Hiệp sĩ đầu tiên đọc vào ở vị trí đầu tiên của mảng. Hiệp sĩ cuối cùng được đọc ở vị trí cuối cùng của mảng. Hiệp sĩ ở vị trí cuối sẽ là Hiệp sĩ giao tranh với các đối thủ và cũng là Hiệp sĩ nhặt được vật phẩm. Khi Hiệp sĩ này hi sinh, Hiệp sĩ liền kề đứng trước Hiệp sĩ này sẽ là Hiệp sĩ tiếp theo nhặt vật phẩm và giao tranh. Quá trình này tiếp tục lặp lại đến khi gặp mảng không còn Hiệp sĩ nào. Khai báo của phương thức khởi tạo:

```
ArmyKnights(const string & file_armyknights);
```

- Phương thức hủy (Destructor) cần thu hồi các vùng nhớ được cấp phát động trong Heap.
- Phương thức fight mô tả quá trình Đội quân Hiệp sĩ giao đấu với Đối thủ opponent và thực hiện các cập nhật lên Đội quân Hiệp sĩ. Method trả về **false** nếu Hiệp sĩ cuối cùng hy sinh sau khi giao đấu, ngược lại trả về **true**.

```
bool fight(BaseOpponent * opponent);
```

- Method **adventure** nhận vào một con trỏ đến đối tượng Events, đây là một class chứa các sự kiện trong hành trình. Method này mô tả quá trình Đội qua đi qua từng sự kiện. Cuối mỗi sự kiện, thông tin của Đội quân cần được in ra sử dụng phương thức printInfo() của class ArmyKnights. Method trả về **true** nếu Đội quân Hiệp sĩ đánh bại Ultimecia, ngược lại method trả về **false**. Khai báo của method như sau:

```
bool adventure(Events * events);
```

- Method **count** trả về số lượng Hiệp sĩ hiện tại trong Đội quân Hiệp sĩ. Lưu ý rằng nếu một Hiệp sĩ hi sinh thì số lượng Hiệp sĩ sẽ bị giảm đi 1. Khai báo của method như sau:

```
int count() const;
```

tức là fight chỉ có tác dụng cho hiệp sĩ cuối lên đấu nếu chết thì hiệp sĩ kế tiếp lên và thay đổiáoos hiệp sĩ,...

con trỏ đến đối tượng Events chứ ko phải eventsArray

do đối tượng Events cấp phát trên Heap nên cũng có 1 con trỏ trỏ đến Events, cũng là con trỏ cần truyền vào ???

dùng mảng heap hay dslk đơn mà phải dùng con trỏ như này ???

- Method **lastKnight** trả về con trỏ đến đối tượng Hiệp sĩ cuối cùng trong Đội quân Hiệp sĩ. Nếu Đội quân không còn Hiệp sĩ nào thì trả về giá trị **NULL**. Khai báo của method như sau:

```
BaseKnight * lastKnight() const;
```

- Các method **hasPaladinShield**, **hasLancelotSpear**, **hasGuinevereHair**, **hasExcaliburSword** trả về **true** nếu Đội quân Hiệp sĩ lần lượt có các báu vật tương ứng: Khiên của Paladin, Ngọn giáo của Lancelot, Sợi tóc của Guinevere và Gươm Excalibur. Ngược lại, giá trị trả về là **false**. Các khai báo cho các method này như sau:

```
bool hasPaladinShield() const;
bool hasLancelotSpear() const;
bool hasGuinevereHair() const;
bool hasExcaliburSword() const;
```

5.4 Vật phẩm

Class **BaseItem** là một abstract class biểu diễn thông tin cho 1 vật phẩm. Class có hai public pure virtual method là:

có lẽ là bỏ hiệp sĩ vô 1 mảng cấp phát động hoặc 1 dslk đơn

- **canUse**

```
virtual bool canUse(BaseKnight * knight) = 0;
```

  Method trả về `true` nếu knight có thể dùng được vật phẩm này, ngược lại trả về `false`.

- **use**

```
virtual void use(BaseKnight * knight) = 0;
```

  Method thực hiện tác động lên Hiệp sĩ biểu diễn bởi knight nhằm thay đổi các thông số của Hiệp sĩ cho phù hợp với tác dụng của vật phẩm.

Các class Antidote, PhoenixDownI, PhoenixDownII, PhoenixDownIII, PhoenixDownIV lần lượt biểu diễn cho các vật phẩm: thuốc giải Antidote, nước mắt phượng hoàng PhoenixDown Loại I, II, III, IV. Các class này kế thừa từ class BaseItem và cần phải định nghĩa lại 2 method canUse và use sao cho phù hợp. Đối với class Antidote, sinh viên có thể định nghĩa thêm thuộc tính cho Hiệp sĩ để biểu diễn trạng thái đang trúng độc.

5.5 Đối thủ

Class **BaseOpponent** là một abstract class biểu diễn thông tin cho 1 đối thủ.

Các class **MadBear**, **Bandit**, **LordLupin**, **Elf**, **Troll**, **Tornbery**, **QueenOfCards**, **NinaDeRings**, **DurianGarden**, **OmegaWeapon**, **Hades** lần lượt là các class biểu diễn cho các đối thủ ở các sự kiện từ 1 đến 11. Các class này kế thừa từ class **BaseOpponent**. Sinh viên tự đề xuất các phương thức trong các class này để khi các đối tượng Hiệp sĩ thực hiện phương thức **fight** thì thông tin của Hiệp sĩ được cập nhật cho phù hợp.

5.6 Túi đồ

Mỗi Hiệp sĩ sẽ mang một túi đồ để chứa các vật phẩm chuẩn bị trước khi bắt đầu hành trình cũng như vật phẩm mà hiệp sĩ nhặt được trong hành trình. Túi đồ được hiện thực bằng một danh sách liên kết đơn. Khi nhặt được một vật phẩm, vật phẩm này sẽ được đặt vào đầu của túi đồ (cũng là đầu của danh sách liên kết đơn). Khi cần tìm một vật phẩm, hiệp sĩ sẽ đi tìm từ đầu đến cuối của túi đồ. Khi muốn sử dụng một vật phẩm, hiệp sĩ sẽ trao đổi vật phẩm cần sử dụng tìm thấy với vật phẩm ở đầu của túi đồ, sau đó xóa vật phẩm cần sử dụng (lúc này ở đầu túi đồ) ra khỏi túi đồ.

Bên cạnh đó, mỗi loại hiệp sĩ sử dụng một túi đồ khác nhau:

- Để tiện cho việc chiến đấu, mỗi Hiệp sĩ không được mang quá nhiều vật phẩm. Túi đồ của Hiệp sĩ Rồng sẽ chứa tối đa 14 vật phẩm, số lượng tối đa tương tự cho Hiệp sĩ LanceLot và Hiệp sĩ Bàn Tròn thông thường lần lượt là 16 và 19 vật phẩm. Khi túi đồ của một Hiệp sĩ đã có số lượng vật phẩm tối đa cho phép, nếu Hiệp sĩ nhặt được một vật phẩm, Hiệp sĩ sẽ đưa nó cho Hiệp sĩ phía trước. Quá trình này tiếp tục đến khi có Hiệp sĩ có thể bỏ được vật phẩm vào túi đồ của họ. Nếu Hiệp sĩ đầu tiên cũng không bỏ được vật phẩm vào túi đồ thì vật phẩm này sẽ bị bỏ qua.
- Hiệp sĩ Rồng do không bị ảnh hưởng bởi độc nên túi đồ của Hiệp sĩ Rồng không chứa thuốc giải antidote. Nếu Hiệp sĩ nhặt được thuốc giải antidote, Hiệp sĩ Rồng sẽ truyền cho Hiệp sĩ phía trước mình và quá trình này tiếp tục như các truyền vật phẩm ở trên.
- Túi đồ của Paladin được Merlin phù phép với phép thuật không gian nên không bị giới hạn số lượng vật phẩm.

Class **BaseBag** là một class biểu diễn Túi đồ. class có 1 thuộc tính **knight** với kiểu là **BaseKnight*** biểu diễn Hiệp sĩ đang sở hữu túi đồ này. Bên cạnh đó, class có các public method là:

- **insertFirst**

```
virtual bool insertFirst(BaseItem * item);
```

tại sao dùng virtual ở đây ???

Method trả về `true` nếu vật phẩm có thể thêm vào túi đồ, ngược lại trả về **false**. Cách thức thêm vật phẩm vào túi đồ là thêm vật phẩm vào đầu danh sách liên kết.

- **get**

```
virtual BaseItem * get(ItemType itemType);
```

Method trả về con trỏ đến một đối tượng biểu diễn cho vật phẩm đầu tiên trong túi có cùng kiểu với **itemType**. Nếu không có vật phẩm cùng kiểu thì trả về giá trị **NULL**. Khi có vật phẩm cùng kiểu itemType thì vật phẩm sẽ được sử dụng, khi đó cần phải xóa vật phẩm khỏi túi đồ (là một danh sách liên kết). Cách thức xóa vật phẩm là trao đổi vật phẩm với vật phẩm ở đầu danh sách liên kết, sau đó xóa, node ở đầu danh sách liên kết. Cách thức tìm vật phẩm là tìm từ đầu danh sách liên kết đến cuối danh sách liên kết.

- toString()

```
virtual string toString() const;
```

Method trả về chuỗi (string) biểu diễn thông tin cho đối tượng BaseItem. Chuỗi biểu diễn có định dạng như sau:

```
Bag[count=<c>;<list_items>]
```

Trong đó:

  - <c>: là số lượng item hiện tại mà túi đồ đang có.
  - <list_items>: là một chuỗi biểu diễn các vật phẩm từ đầu đến cuối của danh sách liên kết, mỗi vật phẩm được biểu diễn bằng tên kiểu của vật phẩm, các vật phẩm được ngăn cách nhau bởi 1 dấu phẩy. Tên kiểu của các vật phẩm Antidote, PhoenixDownI, PhoenixDownII, PhoenixDownIII, PhoenixDownIV lần lượt là Antidote, PhoenixI, PhoenixII, PhoenixIII, PhoenixIV.

Ví dụ về chuỗi biểu diễn túi đồ có 3 vật phẩm theo thứ tự từ đầu đến cuối danh sách lần lượt là Antidote, PhoenixDownIV, PhoenixDownI.

```
Bag[count=3;Antidote,PhoenixIV,PhoenixI]
```

Sinh viên tự đề xuất các class kế thừa từ class BaseBag và biểu diễn các loại túi đồ khác nhau của từng loại Hiệp sĩ. Phương thức khởi tạo (Constructor) của các túi đồ này nhận đầu vào là 3 tham số. Tham số đầu tiên thuộc kiểu **BaseKnight*** biểu diễn cho Hiệp sĩ đang sở hữu túi đồ này. Hai tham số tiếp theo đều kiểu nguyên lần lượt là số phoenixdownI và số antidote ban đầu Hiệp sĩ mang theo. Gọi 2 số này lần lượt là a và b. Nếu $a > 0$ thì túi đồ được khởi tạo gồm có a Node đầu tiên là PhoenixDown loại I. Tiếp theo, nếu $b > 0$ thì túi đồ được thêm vào

b Node Antidote ở đầu danh sách liên kết đơn. Lưu ý khi thêm vào danh sách thì luôn thêm vào ở đầu.

5.7 class KnightAdventure

Hiện thực class **KnightAdventure** có các method:

- **loadArmyKnights** nhận vào 1 tham số kiểu string chứa file mô tả thông tin cho Đội quân Hiệp sĩ. Khởi tạo đối tượng ArmyKnights (là một thuộc tính của class) từ file được cung cấp.
- **loadEvents** nhận vào 1 tham số kiểu string chứa file mô tả thông tin cho các sự kiện. Khởi tạo đối tượng Events (là một thuộc tính của class) từ file được cung cấp.
- **run** không có tham số, mô tả quá trình Đội quân Hiệp sĩ đi qua các sự kiện. Cuối mỗi sự kiện cần in ra thông tin của Đội quân sử dụng printInfo() của class ArmyKnights. Sau sự kiện cuối cùng, gọi hàm **printResult** với thông số truyền vào là **true** nếu Đội quân Hiệp sĩ chiến thắng Ultimecia, hoặc **false** nếu Đội quân bị bại trận.

cái in này sẽ làm trong adventure của army

6 Yêu cầu ĐÃ ĐỌC 6. YÊU CẦU

Để hoàn thành bài tập lớn này, sinh viên phải:

1. Đọc toàn bộ tập tin mô tả này.
2. Tải xuống tập tin initial.zip và giải nén nó. Sau khi giải nén, sinh viên sẽ nhận được các tập tin: main.cpp, main.h, knight2.h, knight2.cpp, và các file dữ liệu đọc mẫu. Sinh viên sẽ chỉ nộp 2 tập tin là knight2.h và knight2.cpp nên không được sửa đổi tập tin main.h khi chạy thử chương trình.
3. Testcases ví dụ cùng kết quả chạy được đăng ở nơi nộp bài. Sinh viên tự chạy thử ví dụ này, nếu ra kết quả khác với Nơi nộp bài thì sinh viên đăng câu hỏi cùng kết quả từng bước chạy trên Diễn đàn.
4. Sinh viên sử dụng câu lệnh sau để biên dịch:
   g++ -o main main.cpp knight2.cpp -I . -std=c++11
   Sinh viên sử dụng câu lệnh sau để chạy chương trình:
   ./main tc1_armyknights tc1_events
   Các câu lệnh trên được dùng trong command prompt/terminal để biên dịch và chạy chương trình. Nếu sinh viên dùng IDE để chạy chương trình, sinh viên cần chú ý: thêm

đầy đủ các tập tin vào project/workspace của IDE; thay đổi lệnh biên dịch của IDE cho phù hợp. IDE thường cung cấp các nút (button) cho việc biên dịch (Build) và chạy chương trình (Run). Khi nhấn Build IDE sẽ chạy một câu lệnh biên dịch tương ứng, thông thường câu lệnh chỉ biên dịch file main.cpp. Sinh viên cần tìm cách cấu hình trên IDE để thay đổi lệnh biên dịch: thêm file knight2.cpp, thêm option -std=c++11, -I .

5. Chương trình sẽ được chấm trên nền tảng Unix. Nền tảng chấm và trình biên dịch của sinh viên có thể khác với nơi chấm thực tế. Nơi nộp bài trên BKeL được cố gắng cài đặt để giống với nơi chấm thực tế. Sinh viên phải chạy thử chương trình trên nơi nộp bài và phải sửa tất cả các lỗi xảy ra ở nơi nộp bài BKeL để có đúng kết quả khi chấm thực tế.
6. Sửa đổi các file knight2.h, knight2.cpp để hoàn thành bài tập lớn này và đảm bảo các yêu cầu sau:
   - Hiện thực các class như mô tả. Sinh viên có thể viết thêm các thuộc tính và các phương thức khác trong các class nếu thấy cần thiết. Tất cả các phương thức trong mô tả này đều phải được hiện thực để việc biên dịch được thực hiện thành công. Nếu sinh viên chưa thể hiện thực được phương thức nào, hãy cung cấp một hiện thực rỗng cho phương thức đó.
   - Cuối mỗi sự kiện, sinh viên cần in ra thông tin của Đội quân sử dụng phương thức **printInfo()** được viết sẵn trong class ArmyKights.
   - Các vùng nhớ được cấp phát động trong Heap phải được thu hồi trước khi chương trình kết thúc.
   - Chỉ có 1 lệnh **include** trong tập tin knight2.h là **#include "main.h"** và một include trong tập tin knight2.cpp là **#include "knight2.h"**. Ngoài ra, không cho phép có một **#include** nào khác trong các tập tin này.
7. Sinh viên được khuyến khích viết thêm các hàm để hoàn thành BTL này.

7 Nộp bài

Sinh viên chỉ nộp 2 tập tin: knight2.h và knight2.cpp, trước thời hạn được đưa ra trong đường dẫn "Assignment 2 - Submission". Có một số testcase đơn giản được sử dụng để kiểm tra bài làm của sinh viên nhằm đảm bảo rằng kết quả của sinh viên có thể biên dịch và chạy được. Sinh viên có thể nộp bài bao nhiêu lần tùy ý nhưng chỉ có bài nộp cuối cùng được tính điểm. Vì hệ thống không thể chịu tải khi quá nhiều sinh viên nộp bài cùng một lúc, vì vậy sinh viên nên nộp bài càng sớm càng tốt. Sinh viên sẽ tự chịu rủi ro nếu nộp bài sát hạn chót. Khi quá thời hạn nộp bài, hệ thống sẽ đóng nên sinh viên sẽ không thể nộp nữa. Bài nộp qua các phương

thức khác đều không được chấp nhận.

8 Một số quy định khác

- Sinh viên phải tự mình hoàn thành bài tập lớn này và phải ngăn không cho người khác đánh cắp kết quả của mình. Nếu không, sinh viên sẽ bị xử lý theo quy định của trường vì gian lận.
- Mọi quyết định của giảng viên phụ trách bài tập lớn là quyết định cuối cùng.
- Sinh viên không được cung cấp testcase sau khi chấm bài, sinh viên sẽ được cung cấp phân bố điểm của BTL.
- Nội dung Bài tập lớn sẽ được Harmony với một câu hỏi trong bài kiểm tra với nội dung tương tự.

9 Gian lận

Bài tập lớn phải được sinh viên TỰ LÀM. Sinh viên sẽ bị coi là gian lận nếu:

- Có sự giống nhau bất thường giữa mã nguồn của các bài nộp. Trong trường hợp này, TẤT CẢ các bài nộp đều bị coi là gian lận. Do vậy sinh viên phải bảo vệ mã nguồn bài tập lớn của mình.
- Bải của sinh viên bị sinh viên khác nộp lên.
- Sinh viên không hiểu mã nguồn do chính mình viết, trừ những phần mã được cung cấp sẵn trong chương trình khởi tạo. Sinh viên có thể tham khảo từ bất kỳ nguồn tài liệu nào, tuy nhiên phải đảm bảo rằng mình hiểu rõ ý nghĩa của tất cả những dòng lệnh mà mình viết. Trong trường hợp không hiểu rõ mã nguồn của nơi mình tham khảo, sinh viên được đặc biệt cảnh báo là KHÔNG ĐƯỢC sử dụng mã nguồn này; thay vào đó nên sử dụng những gì đã được học để viết chương trình.
- Nộp nhầm bài của sinh viên khác trên tài khoản cá nhân của mình.
- Sử dụng code từ ChatGPT.

Trong trường hợp bị kết luận là gian lận, sinh viên sẽ bị điểm 0 cho toàn bộ môn học (không chỉ bài tập lớn).

KHÔNG CHẤP NHẬN BẤT KỲ GIẢI THÍCH NÀO VÀ KHÔNG CÓ BẤT KỲ NGOẠI LỆ NÀO!

Sau mỗi bài tập lớn được nộp, sẽ có một số sinh viên được gọi phỏng vấn ngẫu nhiên để chứng minh rằng bài tập lớn vừa được nộp là do chính mình làm.