

Bài Tập Lớn 2

THANH GƯƠM TRONG ĐÁ

Phiên bản 2.2

1. Giới thiệu



Ultimacia, nữ phù thủy 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, Ultimacia trồng một bông hồng sau vườn. 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. Nếu có một hiệp sĩ nào chiến thắng được Ultimacia trước khi cánh hồng cuối cùng chạm đất, nàng sẽ huỷ bỏ quyết định của mình và nhân loại sẽ được giải cứu khỏi thảm họa diệt vong.

Muốn đánh bại Ultimacia, 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 lâu đài khác nhau của nước Anh rộng lớn và cổ kính.

Ngay cả khi được rút khỏi đá, gươm Excalibur chỉ có thể được sử dụng bởi một hiệp sĩ Bàn Tròn. Vì vậy, một lần nữa các hiệp sĩ Bàn Tròn lại đứng trước một nhiệm vụ mới đầy thử thách và vinh quang: tìm lại thanh gươm trong đá, chiến thắng Ultimacia và giải cứu nhân loại. Liệu thanh gươm trong đá có được rút ra trước khi cánh hồng cuối cùng rơi xuống, liệu Ultimacia có từ bỏ ý định cực đoan trong đau khổ của mình. Tất cả sẽ được giải đáp trong bài tập lớn này.

2. Yêu cầu

Trong bài tập lớn này, sinh viên sẽ được cung cấp một file chứa dữ liệu nhập, bao gồm thông tin về chế độ cài đặt ban đầu của chương trình, thông tin về hiệp sĩ Bàn Tròn đi tìm thanh gươm trong đá và các lâu đài phải qua. Sinh viên sẽ hiện thực một hàm đặc biệt là hàm *walkthrough*. Chi tiết về hàm *walkthrough* sẽ được trình bày rõ hơn ở phần 4.

3. Dữ liệu nhập

Dữ liệu nhập của chương trình được chứa trong file mang tên `input.txt`. File này sẽ chứa các thông tin như sau:

```
0 12
172 4 0 100
95 96 97 0 98 99 0
```

The diagram shows three lines of input data. The first line '0 12' is linked to a box labeled 'chế độ cài đặt ban đầu của chương trình'. The second line '172 4 0 100' is linked to a box labeled 'thông tin về hiệp sĩ Bàn Tròn'. The third line '95 96 97 0 98 99 0' is linked to a box labeled 'thông tin về các lâu đài hiệp sĩ sẽ qua'.

Như vậy file nhập sẽ bao gồm ít nhất ba dòng. Dòng đầu tiên sẽ mô tả thông tin cài đặt chương trình, có định dạng như sau

mode petals

Trong đó:

mode: các chế độ thực thi của chương trình. Nếu $mode = 0$, chương trình thực thi ở chế độ bình thường. Các giải thích trong phần 4 từ mục a đến mục u là dành cho $mode = 0$. Các $mode$ khác 0 sẽ được giải thích ở mục v và mục x của phần 4.

petal: số cánh hoa trên bông hoa hồng ở vườn nhà Ultimecia, là một số nguyên từ 0 đến 99

Dòng thứ hai sẽ mô tả thông tin về hiệp sĩ Bàn Tròn, có định dạng như sau:

HP level antidote gil

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

antidote: Số thuốc giải độc antidote mà hiệp sĩ mang theo, là một số nguyên có giá trị từ 0 đến 99

gil: Số tiền mà hiệp sĩ mang theo, là một số nguyên có giá trị từ 1 đến 999.

Kể từ dòng thứ ba của file nhập sẽ mô tả hành trình đi tìm thanh gươm trong đá. Hành trình sẽ đi qua nhiều lâu đài, mỗi lâu đài bao gồm nhiều sự kiện. 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 [Bảng 1](#). Các sự kiện ở các lâu đài ngăn cách nhau bởi số nguyên 0. Số sự kiện và số lâu đài là không cố định, có thể thay đổi tùy theo test case. Một sự kiện có thể xảy ra nhiều lần. Sẽ có tối đa 100 sự kiện xảy ra. Mỗi lâu đài có tối đa 20 sự kiện. Nếu số sự kiện nhiều, các sự kiện có thể trình bày thành nhiều dòng. Trong mỗi lâu đài, mỗi sự kiện sẽ được đánh chỉ số bắt đầu từ 1, khi qua lâu đài mới, chỉ số sự kiện sẽ được đánh lại từ 1.

Ví dụ 1: Với dữ liệu nhập là

0 12

172 4 0 100

95 96 97 0 98 99 0

thì chương trình sẽ thực thi ở mode 0, có tất cả 12 cánh hoa hồng. Hiệp sĩ Bàn Tròn đi tìm thanh gươm trong đá sẽ có *HP* là 172, *level* là 4, hiệp sĩ không có *antidote*, và có 100 *gil*. Mặt khác, *MaxHP* của hiệp sĩ cũng là 172 (nghĩa là *HP* của hiệp sĩ sẽ không vượt quá 172, trừ khi *MaxHP* thay đổi).

Trên đường đi, hiệp sĩ sẽ đi qua hai lâu đài, có các sự kiện xảy ra như sau:

Lâu đài 1

Sự kiện 1: gặp khiên của Paladin

Sự kiện 2: gặp giáo của Lancelot

Sự kiện 3: gặp sợi tóc của Guinevere

Lâu đài 2

Sự kiện 1: gặp gươm Excalibur

Sự kiện 2: gặp Ultimecia

4. Hiện thực chương trình

Sinh viên sẽ hiện thực một hàm *walkthrough* có prototype như sau:

*report** *walkthrough (knight& theKnight, castle arrCastle[], int nCastle, int mode, int nPetal)*

Trong đó *report* là một *struct* được khai báo như sau:

```
struct report  
{  
    int nPetal;  
    int nWin;  
    int nLose;  
}
```

; thành phần *nPetal* sẽ là số cánh hoa hồng còn lại sau khi hiệp sĩ chiến thắng Ultimecia, *nWin* và *nLose* là số trận chiến thắng và chiến bại tương ứng của hiệp sĩ trong suốt quá trình đi tìm thanh gươm trong đá.

Thông tin về hiệp sĩ được truyền qua biến tham khảo *theKnight* thuộc cấu trúc *knight* được khai báo như sau:

```
struct knight  
{  
    int HP;  
    int level;  
    int antidote;  
    int gil;  
}
```

Các thành phần *HP*, *level*, *antidote* và *gil* của *struct knight* tương ứng với các thông số *HP*, *level*, *antidote* và *gil* của hiệp sĩ.

Các thông tin về lâu đài sẽ được truyền thông qua biến *arrCastle*, là một dãy các cấu trúc *castle* được định nghĩa như sau:

```
struct castle  
{  
    int arrEvent[20];
```

```
int nEvent;
```

```
}
```

Biến *nCastle* cho biết số lâu đài hiệp sĩ phải qua, các biến *mode* và *nPetal* cho biết thông tin về mode thực thi chương trình và số cánh hoa hồng ban đầu. Hiệp sĩ sẽ lần lượt đi qua các sự kiện chứa trong dãy *arrCastle* (sẽ được chứa theo đúng thứ tự các sự kiện được mô tả trong file *input.txt*). Mỗi khi hiệp sĩ đi qua một sự kiện, một cánh hoa sẽ rơi xuống. Nếu số cánh hoa là 0 mà hiệp sĩ vẫn chưa thắng được Ultimecia, hàm *walkthrough* sẽ chấm dứt và trả về giá trị NULL.

Ví dụ 2: Với dữ liệu nhập là

0 1

172 4 0 100

95 96 97 0 98 99 0

Ở sự kiện 1 tại lâu đài thứ 1, hiệp sĩ nhặt được chiếc khiên của Paladin. Tuy nhiên do chỉ có 1 cánh hoa hồng, nên sau khi hiệp sĩ nhặt được khiên, cánh hoa hồng này rơi xuống, hàm *walkthrough* sẽ trả về giá trị NULL.

Trong quá trình đi tìm gươm, các thông tin về hiệp sĩ sẽ thay đổi dựa trên các sự kiện mà hiệp sĩ gặp trên đường đi, cụ thể như sau:

a) Nếu hiệp sĩ gặp Ultimecia và có gươm Excalibur, hiệp sĩ sẽ đánh thắng Ultimecia, hàm *walkthrough* sẽ chấm dứt và trả về một cấu trúc *report* tương ứng. Ngược lại nếu không có gươm Excalibur, hiệp sĩ sẽ đánh thua Ultimecia, HP của hiệp sĩ sẽ giảm như sau: $HP = HP/3$ (chỉ tính phần nguyên). Nếu HP của hiệp sĩ nhỏ hơn 3, HP của hiệp sĩ sẽ được giảm còn 1.

b) Nếu hiệp sĩ gặp gươm Excalibur mà 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 này và tiếp tục cuộc hành trình để tìm Ultimecia.

c) Khi qua khỏi một lâu đài, level của hiệp tăng 1 đơn vị, nhưng không được tăng quá 10. Nếu đã đi qua hết tất cả các lâu đài, hiệp sĩ sẽ quay lại bắt đầu từ đầu. Mỗi khi được tăng 1 level, *MaxHP* của hiệp sĩ sẽ được tăng lên 100 đơn vị, tuy nhiên *MaxHP* không được phép vượt quá 999.

Chú ý: Trong tất cả các test case, các sự kiện 95,96,97,98 và 99 (nhặt được 3 báu vật, gặp gươm Excalibur và gặp Ultimecia) sẽ chắc chắn xảy ra.

Ví dụ 3: Với dữ liệu nhập là

0 12

172 4 0 100

95 96 97 0 98 99 0

Sau 3 sự kiện ở lâu đài thứ 1, hiệp sĩ nhặt được đầy đủ ba báu vật. Sau khi ra khỏi lâu đài 1, *level* của hiệp sĩ sẽ được tăng lên 1 (*level* = 5) và số cánh hoa hồng sẽ còn $12-3=9$. Sang sự kiện 1 ở lâu đài 2, hiệp sĩ rút được Excalibur do đã có đủ 3 báu vật. Đến sự kiện 2 ở lâu đài 2, hiệp sĩ gặp Ultimecia và chiến thắng, lúc này sẽ còn lại $9-2=7$ cánh hoa hồng. Như vậy thông tin của biến *theKnight* khi chiến thắng sẽ là {172,5,0,100}. Kết quả trả về của hàm *walkthrough* sẽ là một con trỏ trỏ đến cấu trúc có giá trị là {7,1,0}; tức còn 7 cánh hoa, thắng 1 trận (thắng Ultimecia) và thua 0 trận. (Chú ý rằng lúc này *MaxHP* của hiệp sĩ đã được tăng lên 272, nhưng *HP* của hiệp sĩ vẫn là 172).

Ví dụ 4: Với dữ liệu nhập là

0 8

172 4 0 100

96 98 99 0 97 95 0

Ở sự kiện 1, hiệp sĩ nhặt được giáo của Lancelot, đến sự kiện 2 gặp được gươm Excalibur nhưng không thể rút ra khỏi đá do chưa đủ ba báu vật. Đến sự kiện thứ 3, hiệp sĩ đánh thua Ultimecia, *HP* của hiệp sĩ sẽ còn 57. Sau khi ra khỏi lâu đài 1, *level* của hiệp sĩ sẽ được tăng lên 1 (*level* = 5) và số cánh hoa hồng sẽ còn $8-3=5$. Sang lâu đài thứ 2, hiệp sĩ nhặt được 2 báu vật còn lại, số cánh hoa hồng còn lại $5-2=3$ cánh. Sau khi ra khỏi lâu đài thứ 2, *level* của hiệp sĩ tiếp tục tăng lên 1 (*level* = 6). Do đã đi đủ các lâu đài, hiệp sĩ sẽ quay lại đi từ đầu, bắt đầu từ lâu đài 1. Lần này hiệp sĩ rút được gươm Excalipur và chiến thắng Ultimecia, vừa lúc cánh hoa cuối cùng rơi. Như vậy thông tin của biến *theKnight* khi chiến thắng sẽ là {57,6,0,100}. Kết quả trả về của hàm *walkthrough* sẽ là một con trỏ trỏ đến cấu trúc có giá trị là {0,1,1}; tức còn 0 cánh hoa, thắng 1 trận và thua 1 trận.

d) Nếu gặp sự kiện có mã từ 1 đến 5, hiệp sĩ phải giao tranh với đối thủ tương ứng. Mỗi đối thủ cũng sẽ có đẳng cấp *levelO* riêng. Nếu gặp đối thủ ở sự kiện thứ *i*, *levelO* tương ứng của đối thủ sẽ được tính như sau:

$$b = i \% 10 \quad (1)$$

$$levelO = i > 6 ? (b > 5 ? b : 5) : b \quad (2)$$

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 [Bảng 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:

$$HP = HP - damage \quad (3)$$

damage sẽ được tính như sau:

$$damage = baseDamage * levelO * 10 \quad (4)$$

trong đó *baseDamage* sẽ tùy thuộc vào đối thủ, được mô tả ở [Bảng 3](#).

Nếu hiệp sĩ chưa nhặt được gươm Excalibur thì sẽ luôn giao tranh thua tất cả các đối thủ bất chấp level là bao nhiêu.

Lưu ý rằng *HP* sẽ luôn là số nguyên khi tính bằng công thức (3). Nếu *HP* nhỏ hơn hoặc bằng 0 sau khi tính bằng công thức (3), hàm *walkthrough* sẽ tự động gọi hàm *callPhoenix* (xem thêm mô tả ở phần 5 về hàm *callPhoenix*) để gọi phượng hoàng đến hồi sinh hiệp sĩ. Khi gọi phượng hoàng, số tiền của hiệp sĩ sẽ bị trừ đi 100 gil. Nếu số tiền của hiệp sĩ nhỏ hơn 100 (kể cả khi *gil* = 0) trước khi gọi phượng hoàng, số tiền của hiệp sĩ sẽ được giảm thành 0. Khi hồi sinh, *HP* của hiệp sĩ sẽ bằng $1/2 \text{ MaxHP}$ (chỉ tính phần nguyên).

Ví dụ 5: Với dữ liệu nhập là

0 12

4 1 0 100

95 0 96 97 98 2 99 0

Sau khi qua khỏi lâu đài 1, hiệp sĩ đã nhặt được chiếc khiên của Paladin, *level* được tăng lên 2, do đó *MaxHP* sẽ được tăng lên 104. Đến lâu đài 2, hiệp sĩ nhặt tiếp 2 báu vật còn lại, rút được gươm Excalibur và gặp MoonBringer có *levelO* là 4. Hiệp sĩ đánh thua và *HP* bị giảm xuống còn $4 - 1.5 * 4 * 10 < 0$. Hiệp sĩ sẽ gọi phượng hoàng và hồi sinh với $HP = 104/2 = 52$, *gil* còn $100 - 100 = 0$. Sau đó hiệp sĩ đánh thắng Utimecia và hàm *walkthrough* kết thúc. Thông tin của hiệp sĩ là {52,2,0,0}, thông tin report là {6,1,1}.

e) Nếu gặp ma Tornbery, hiệp sĩ sẽ giao tranh với Tornbery. Cách giao tranh tương tự như mô tả trong mục d. 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 trong 5 sự kiện tiếp theo. Khi đang bị trúng độc và phải giao tranh với các đối thủ có mã từ 1-5 hoặc với Utimecia, nếu đánh thắng hiệp sĩ cũng bị giảm HP giống như khi thua trận.

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, khi đó chỉ số *antidote* của hiệp sĩ bị giảm đi 1. Nếu *HP* của hiệp sĩ giảm xuống bằng hoặc dưới 0 khi hiệp sĩ bị trúng độc, hiệp sĩ sẽ gọi phượng hoàng và được hồi sinh như mô tả ở mục d. Khi hồi sinh, hiệp sĩ sẽ tự động được giải độc. Tornbery sẽ không đánh với hiệp sĩ đang bị trúng độc.

Bảng 1 – Các sự kiện trên hành trình tìm thanh gươm trong đá

Mã sự kiện	Ý nghĩa
1	Gặp gấu MadBear
2	Gặp nữ chiến binh Amazon MoonBringer

3	Gặp yêu tinh Elf
4	Gặp chiến binh Saxon
5	Gặp quỷ khổng lồ Troll
6	Gặp ma Tornbery
7	Gặp nữ hoàng cờ bạc Queen of Cards
8	Gặp lái buôn vui tính Nina de Rings
9	Lạc vào Vườn Sầu Riêng
10	Nhặt được thuốc giải Antidote
11	Gặp thần Odin
12	Gặp phù thủy Merlin
13	Gặp quái vật Omega Weapon
14	Gặp Thần Chết Hades
15	Nhặt được Scarlet Hakama
16	Gặp cánh cổng thử thách LockedDoor
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

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

Đối thủ	gil
MadBear	100
MoonBringer	150
Elf	450
Saxon	650
Troll	850

Ví dụ 6: Với dữ liệu nhập là

0 12

172 1 0 100

95 0 96 97 98 6 0 1 99 0

Sau khi qua khỏi lâu đài 1, hiệp sĩ đã nhặt được chiếc khiên của Paladin, *level* =2, *MaxHP* = 272. Đến lâu đài 2, hiệp sĩ nhặt tiếp 2 báu vật còn lại, rút được gươm Excalibur và gặp Tornbery có *level* 0 là 4. Hiệp sĩ đánh thua và trúng độc. Sau đó hiệp sĩ ra khỏi lâu đài 2, *level* = 3, *MaxHP* = 372. Sau đó ở lâu đài 3 hiệp sĩ đánh thắng MadBear, được thêm 100

gil nhưng *HP* vẫn giảm còn $HP = 172 - 1*1*10 = 162$. Sau đó hiệp sĩ đánh thắng tiếp Ultimecia nhưng *HP* vẫn giảm còn $162/3 = 54$. Hàm *walkthrough* kết thúc. Thông tin của hiệp sĩ là $\{54,3,0,200\}$, thông tin report là $\{5,2,1\}$.

Bảng 3 – Chỉ số baseDamage của các đối thủ

Đối thủ	baseDamage
MadBear	1
MoonBringer	1.5
Elf	4.5
Saxon	6.5
Troll	8.5

f) 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 d. Nếu hiệp sĩ thắng, *gil* của hiệp sĩ sẽ được tăng gấp đôi. Nếu thua, hiệp sĩ sẽ bị giảm một nửa số *gil* (chỉ tính phần nguyên).

g) 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:

- Hiệp sĩ sẽ tiếp tục hành trình và không mua bán gì cả nếu có ít hơn 50 *gil*.
- Nếu hiệp sĩ đang trúng độc, hiệp sĩ sẽ đưa Nina 50 *gil* để được giải độc.
- Nếu hiệp sĩ vẫn còn *gil* sau khi giải độc hoặc không trúng độc, hiệp sĩ sẽ quy đổi *gil* thành *HP* theo tỷ lệ 1:1 cho đến khi hiệp sĩ hết tiền hoặc *HP* của hiệp sĩ đạt được giá trị *MaxHP*.

Ví dụ 6b: Với dữ liệu nhập là

0 12

172 1 0 100

95 96 97 98 1 8 99 0

Sau khi nhặt đủ 3 báu vật và rút được guom Excalibur, hiệp sĩ đánh với MadBear có *levelO* là 5 và thua, $HP = 172 - 1*5*10 = 122$. Sau đó hiệp sĩ gặp Nina, hiệp sĩ sẽ đưa Nina 50 *gil* để phục hồi *HP* thành *MaxHP* là 172 như cũ. Sau đó hiệp sĩ đánh thắng tiếp Ultimecia và hàm *walkthrough* kết thúc. Thông tin của hiệp sĩ là $\{172,1,0,50\}$, thông tin report là $\{5,1,1\}$.

Tuy nhiên, ngay khi gặp Nina, nếu số *HP* và số *gil* của hiệp sĩ tạo thành một cặp số bạn (*friendly number* - http://en.wikipedia.org/wiki/Friendly_number), Nina sẽ không thu tiền khi giải độc và phục hồi *HP* cho hiệp sĩ (ngay cả khi hiệp sĩ có ít hơn 50 *gil*). Ngoài ra Nina sẽ tặng hiệp sĩ thanh kiếm *Lionheart*. Khi có kiếm *Lionheart*, hiệp sĩ đánh thắng tất cả mọi đối thủ, kể cả Ultimecia và Hades (nghĩa là hàm *walkthrough* sẽ kết thúc nếu hiệp sĩ chưa rút được Excalibur nhưng gặp Ultimecia khi đang có *Lionheart*). Tuy nhiên, sau 5 sự kiện,

kiếm Lionheart sẽ tự động bay về với Nina và hiệp sĩ lại tiếp tục hành trình bình thường. Nina không bao giờ xuất hiện quá 1 lần trong 6 sự kiện liên tiếp.

h) Nếu hiệp sĩ lạc vào Vườn Sầu Riêng, hiệp sĩ sẽ được giải độc nếu đang trúng độc, *HP* của hiệp sĩ sẽ phục hồi về *MaxHP*, đồng thời hoa hồng của Ultimecia sẽ tự động mọc thêm 5 cánh, tuy nhiên số cánh hoa hồng không bao giờ vượt quá 99. Khi ra khỏi Vườn Sầu Riêng, cánh hoa hồng vẫn rơi như thường lệ.

i) Nếu hiệp sĩ nhặt được thuốc giải Antidote, chỉ số *antidote* của hiệp sĩ sẽ tăng lên 1. Nếu hiệp sĩ nhặt được Antidote khi đang trúng độc, thuốc giải này sẽ được dùng ngay lập tức.

k) Nếu hiệp sĩ gặp Odin, Odin sẽ theo giúp hiệp sĩ trong 5 sự kiện tiếp theo. Hiệp sĩ có Odin theo giúp sẽ đánh thắng các đối thủ có mã sự kiện từ 1-7 bất chấp *levelO* của đối thủ. Odin không bao giờ xuất hiện quá 1 lần trong vòng 6 sự kiện liên tiếp

l) Nếu hiệp sĩ gặp Merlin, Merlin sẽ làm các công việc theo thứ tự sau:

- Giải độc cho hiệp sĩ nếu hiệp sĩ trúng độc
- Tăng level của hiệp sĩ lên 1 (*MaxHP* của hiệp sĩ cũng được tăng theo tương ứng)
- Phục hồi *HP* của hiệp sĩ thành *MaxHP*

m) Hiệp sĩ có thể gặp quái vật Omega Weapon, là 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 hiệp sĩ phải gọi phượng hoàng hồi sinh để tiếp tục cuộc hành trình. Chỉ có hiệp sĩ ở level 10 đang có Excalibur hoặc Hiệp sĩ Rỗng ở level bất kỳ đang có Lionheart mới có thể đánh thắng Omega Weapon. Trong trường hợp đánh thắng Omega Weapon, level của hiệp sĩ được tăng thành 10 (*MaxHP* sẽ tăng theo tương ứng), *gil* của hiệp sĩ được tăng thành 999. Sau khi bị đánh bại, Omega Weapon sẽ không bao giờ xuất hiện lại (nghĩa là nếu hiệp sĩ có gặp tiếp sự kiện có mã là 13, hiệp sĩ chỉ đơn giản bỏ qua sự kiện này và đi tiếp, trong trường hợp này, hoa hồng vẫn rơi tiếp một cánh).

n) Khi gặp Thần Chết Hades, hiệp sĩ sẽ giao tranh với Hades, cách giao tranh tương tự như mô tả trong mục d. Nếu đánh thua Hades, *HP* của hiệp sĩ sẽ bị giảm về 0 và phải gọi phượng hoàng để hồi sinh. Nếu đánh thắng Hades, Hades sẽ rèn cho hiệp sĩ áo giáp mythril. Hiệp sĩ mặc áo giáp mythril sẽ không bao giờ giảm *HP* cho dù bại trận kể cả khi giao tranh với Hades, Omega Weapon và Ultimecia. Tuy nhiên mặc áo giáp mythril vẫn có thể trúng độc nếu thua Tornbery.

Ví dụ 7: Với dữ liệu nhập là

172 1 0 100

14 95 96 97 98 14 6 99 0

Hiệp sĩ sẽ gặp Hades và đánh thắng, nhờ đó Hades rèn cho hiệp sĩ áo giáp Mythril. Sau đó hiệp sĩ nhặt đủ các báu vật và rút được gươm Excalibur. Hiệp sĩ gặp lại Hades lần thứ 2 và thua, nhưng nhờ đang mặc áo giáp Mythril nên không bị giảm *HP* và không cần gọi phương hoàng. Sau đó hiệp sĩ gặp tiếp Tornbery và thua nên bị trúng độc. Cuối cùng hiệp sĩ gặp Ultimecia và chiến thắng, nhưng do đang trúng độc nên HP của hiệp sĩ vẫn bị giảm còn 57. Hàm *walkthrough* kết thúc, thông tin của hiệp sĩ là {57,1,0,100}, thông tin report là {4,2,2}.

Nếu Odin đang theo giúp hiệp sĩ, Hades sẽ giết Odin ngay lập tức, kể từ lúc này về sau, nếu gặp sự kiện có mã 11, Odin vẫn sẽ không xuất hiện để giúp hiệp sĩ, tuy nhiên cánh hoa hồng vẫn rơi.

Ví dụ 8: Với dữ liệu nhập là

0 12

172 3 0 100

11 14 95 96 97 98 11 1 99 0

Hiệp sĩ gặp Odin ở sự kiện 1, sau đó gặp tiếp Hades ở sự kiện 2. Hades ngay lập tức giết Odin và giao tranh với hiệp sĩ như bình thường. Do hiệp sĩ thắng Hades nên sẽ được rèn một áo giáp mythril. Sau đó hiệp sĩ nhặt các báu vật, rút được gươm Excalibur và gặp MadBear và thua, tuy nhiên nhờ đang mặc áo giáp mythril nên không bị giảm *HP*. Cuối cùng hiệp sĩ đánh thắng Ultimecia và hàm *walkthrough* kết thúc. Thông tin của hiệp sĩ là {172,3,0,100}, thông tin report là {3,2,1}.

Tuy nhiên khi gặp Hades, nếu hiệp sĩ mang trong mình sức mạnh của Tình Yêu Vĩnh Cửu (Eternal Love), sức mạnh của tình yêu sẽ chiến thắng cái chết, Hades sẽ thua bất chấp mọi *levelO* đang có. Hiệp sĩ có Eternal Love trong các trường hợp sau:

- Hiệp sĩ đang có ngọn giáo của Lancelot và sợi tóc của Guinevere nhưng không có Excalibur
- Hiệp sĩ là Arthur và đang có sợi tóc của Guinevere
- Hiệp sĩ là Lancelot và đang có sợi tóc của Guinevere
- Hiệp sĩ là Guinevere và đang có ngọn giáo của Lancelot

o) Scarlet Hakama (SH) là một loại y phục bí truyền của các võ sĩ đạo Nhật Bản. Nếu nhặt được Scarlet Hakama hiệp sĩ sẽ không bị mất *gil* nếu thua Queen of Cards, sẽ không bị Nina thu tiền khi giải độc và đổi *HP*. Nếu vào Vườn Sầu Riêng khi đang có Scarlet Hakama, số cánh hoa hồng sẽ được tăng thành 99.

p) Cổng thử thách LockedDoor là nơi thử thách lòng can đảm của các hiệp sĩ. Nếu gặp LockedDoor ở sự kiện thứ i , hiệp sĩ chỉ vượt qua được LockedDoor nếu *level* của hiệp sĩ lớn hơn $i\%10$. Nếu không vượt qua khỏi LockedDoor hiệp sĩ phải ra khỏi lâu đài và chuyển sang lâu đài kế tiếp, tuy nhiên *level* và *MaxHP* của hiệp sĩ vẫn được tăng lên như khi ra khỏi lâu đài theo cách bình thường.

Ví dụ 9: Với dữ liệu nhập là

0 12

172 2 0 100

95 16 96 97 98 99 0

Ở sự kiện 1, hiệp sĩ nhặt được chiếc khiên của Paladin, tuy nhiên đến sự kiện 2, hiệp sĩ không qua được cổng thử thách và ra khỏi lâu đài, *level* của hiệp sĩ sẽ được tăng lên thành 3. Do không còn lâu đài kế tiếp, hiệp sĩ quay lại đi lại từ đầu. Lần này hiệp sĩ vượt qua được cổng thử thách, nhặt đủ các báu vật và rút được gươm Excalibur. Cuối cùng hiệp sĩ đánh thắng Ultimecia và hàm *walkthrough* kết thúc. Thông tin của hiệp sĩ là $\{172,3,0,100\}$, thông tin report là $\{4,1,0\}$.

q) Nếu *HP* ban đầu của hiệp sĩ là 999, hiệp sĩ chính là vua Arthur tái xuất để giao tranh cùng Ultimecia. Arthur có thể rút được gươm Excalibur bất kỳ lúc nào mà không cần có đủ ba báu vật. Arthur đánh thắng tất cả các đối thủ có mã từ 1-7 ở bất kỳ level nào. Arthur không bị giới hạn về số cánh hoa hồng. Nếu số cánh hoa hồng là 0, Arthur vẫn tiếp tục cuộc hành trình cho đến khi chiến thắng.

Ví dụ 9: Với dữ liệu nhập là

0 1

999 2 0 100

10 98 99 95 96 97 0

Ở sự kiện 1, Arthur nhặt được một viên antidote. Ngay lúc này số cánh hoa hồng đã rơi hết, nhưng Arthur vẫn tiếp tục cuộc hành trình, rút được gươm Excalibur và đánh thắng Ultimecia. Hàm *walkthrough* kết thúc. Thông tin của hiệp sĩ là $\{999,2,1,100\}$, thông tin report là $\{0,1,0\}$.

r) Nếu *HP* ban đầu của hiệp sĩ là 888, hiệp sĩ chính là Lancelot. Lancelot mặc định đã có sẵn ngọn giáo của mình (báu vật có mã 96) và chỉ cần tìm hai báu vật còn lại để rút được Excalibur. Lancelot đánh thắng tất cả các đối thủ có mã từ 1-7 và vượt qua được tất cả mọi cổng thử thách ở bất kỳ level nào.

s) Nếu *HP* ban đầu của hiệp sĩ là 777, hiệp sĩ chính là công chúa Guinevere. Guinevere mặc định đã có sợi tóc của mình (báu vật có mã 97) và chỉ cần tìm hai báu vật còn lại để

rút được Excalibur. Nhờ sắc đẹp của mình, Guinevere được sự ưu ái của các đối thủ nữ. Khi đánh thua MoonBringer và Ultimecia, Guinevere sẽ không bị giảm *HP*, khi đánh thua Queen of Cards, Guinevere sẽ không bị giảm *gil*. Nina sẽ không thu tiền khi giải độc và đổi *HP* cho Guinevere mà còn tặng cho nàng thêm 50 *gil*, tuy nhiên số *gil* của Guinevere không được tăng quá 999.

t) Nếu *HP* ban đầu của hiệp sĩ là một số nguyên tố, hiệp sĩ đó chính là một Paladin. Paladin mặc định đã có chiếc khiên của mình (báu vật có mã 95) và chỉ cần tìm hai báu vật còn lại để rút được Excalibur. Paladin đánh thắng tất cả các đối thủ có mã từ 1-5 ở bất kỳ level nào. Khi thua Tornbery, Paladin không bị trúng độc. Theo một đạo luật cổ được truyền lại từ ngàn xưa, Nina sẽ không thu tiền khi giải độc và đổi *HP* cho Paladin. Nếu Paladin có kiếm Lionheart, thanh kiếm này sẽ ở lại vĩnh viễn với Paladin mà không quay về với Nina sau 5 sự kiện.

u) Trong số các hiệp sĩ Bàn Tròn, có một số hiệp sĩ đặc biệt mang trong mình dòng máu của Rồng, gọi là các Hiệp sĩ Rồng. Một hiệp sĩ là Hiệp sĩ Rồng nếu *HP* ban đầu của hiệp sĩ là tổng của ba số nguyên dương x, y, z sao cho $x^2 + y^2 = z^2$. Lưu ý rằng nếu *HP* ban đầu của hiệp sĩ là 888, hiệp sĩ này không phải là hiệp sĩ Rồng mà là Lancelot. Hiệp sĩ Rồng khi kết hợp với Odin sẽ đánh thắng được Hades (Odin không bị Hades giết trong trường hợp này và sẽ tiếp tục theo giúp hiệp sĩ cho đủ 5 sự kiện). Hiệp sĩ Rồng sẽ không bị trúng độc nếu thua Tornbery và có khả năng vượt qua mọi cổng thử thách với bất kỳ level nào.

v) Nếu thông số cài đặt ban đầu có *Mode* = 1, hiệp sĩ chỉ có thể nhặt các báu vật theo thứ tự được quy định bởi một hàm *hash* (xem thêm phần 5 về hàm *hash*). Báu vật có giá trị *hash* nhỏ hơn phải được nhặt trước báu vật có giá trị *hash* cao hơn. Các báu vật khác nhau sẽ có giá trị *hash* khác nhau.

Ví dụ 10: Với dữ liệu nhập là

1 12

4 1 0 100

95 96 97 98 99 0

Giả sử $\text{hash}(95) = 2$, $\text{hash}(96) = 1$, $\text{hash}(97) = 3$. Ban đầu hiệp sĩ gặp chiếc khiên của Paladin, nhưng do chưa nhặt được ngọn giáo của Lancelot nên hiệp sĩ không nhặt được chiếc khiên này, tiếp đến hiệp sĩ gặp sợi tóc của Guinevere và cũng không nhặt được do chưa có chiếc khiên của Paladin. Tiếp đó hiệp sĩ không rút được thanh Excalibur khỏi đá và đánh thua Ultimecia, *HP* bị giảm còn 1. Sau đó hiệp sĩ ra khỏi lâu đài, *level* = 2, và quay lại từ đầu. Lần này hiệp sĩ lần lượt nhặt được các báu vật, rút được Excalibur và đánh thắng Ultimecia. Hàm *walkthrough* kết thúc. Thông tin của hiệp sĩ là {1,2,0,100}, thông tin report là {2,1,1}.

x) (*bonus*) Nếu thông số cài đặt ban đầu có Mode =2, hiệp sĩ sẽ chọn đường đi tối ưu nhất qua các lâu đài. Đường đi P_1 được xem là tối ưu hơn P_2 nếu như số cánh hoa hồng còn lại sau khi đi theo đường P_1 nhiều hơn đi theo đường P_2 . Test case sẽ đảm bảo chỉ có một đường đi tối ưu. Sự kiện Vườn Sầu Riêng sẽ không xảy ra trong mode 2.

Ví dụ 11: Với dữ liệu nhập là

2 12

172 4 0 100

98 99 0 95 96 97 0

Đường đi tối ưu sẽ là đi qua lâu đài 2 trước rồi mới đến lâu đài 1. Hiệp sĩ sẽ đi theo con đường này. Khi hàm *walkthrough* kết thúc, thông tin của hiệp sĩ là {172,5,0,100}, thông tin report là {7,1,0}.

5. Khởi tạo

Sinh viên download file *Assignment_2.zip* từ trang Web của môn học. Khi giải nén file này, sẽ có được các file sau:

input.txt	Một file input ví dụ.
main.cpp	Chương trình chính
sword.cpp	Chương trình hiện thực bởi sinh viên
defs.h	File định nghĩa các cấu trúc và hàm dùng chung
Assignment_2.pdf	File mô tả nội dung bài tập lớn

File *input.txt* là một file nhập mẫu như được mô tả ở phần 3. File *main.cpp* là chương trình khởi tạo, bao gồm các hàm viết sẵn như sau:

- *main()*: chương trình chính sẽ thực thi
- *readFile()*: hàm đọc file input
- *display()* : hàm xuất dữ liệu ra màn hình.
- *callPhoenix()*: sinh viên phải gọi hàm này khi gọi phượng hoàng hồi sinh hiệp sĩ
- *hash()*: hàm cho biết giá trị hash của mỗi sự kiện. Hàm hash viết sẵn trong file *main.cpp* chỉ có giá trị tham khảo. Khi chấm hàm hash có thể bị thay đổi.

Lưu ý rằng **sinh viên không được phép thay đổi file *main.cpp* và *defs.h* khi hiện thực chương trình.**

Sinh viên sẽ hiện thực hàm *walkthrough* trong file *sword.cpp* với một số yêu cầu như sau:

- Hàm *walkthrough* không được xuất bất kỳ dữ liệu nào ra màn hình khi thực thi.
- Kết quả trả về bởi hàm *walkthrough* phải là con trỏ đến một cấu trúc report được tạo ra bằng lệnh cấp phát động (lệnh *new*).

Để dịch và thực thi chương trình, sinh viên chứa cả 3 files *main.cpp*, *sword.cpp* và *defs.h* trong cùng một thư mục; Sau đó chỉ cần dịch và thực thi **duy nhất** file *main.cpp*. Mọi công việc cần phải làm sẽ được hiện thực trong file *sword.cpp*, tuy nhiên không cần dịch và thực thi file này.

Ví dụ: Để dịch và thực thi chương trình trên môi trường Cygwin, thực thi các lệnh sau:

- `g++ main.cpp`, kết quả ra được file `a.out`
- Tiếp theo chạy lệnh: `./a.out input.txt` với `input` là testcase được đặt chung thư mục với file chứa code.

6. Nộp bài

Thời hạn chót để nộp bài là **23h55 chủ nhật, 07/06/2020**. Sinh viên phải dùng account đã đăng ký trên hệ thống BKeL để nộp bài. KHÔNG nhận bài được gửi qua mail hoặc bất kỳ hình thức nào khác. Bài nộp trễ sẽ KHÔNG được nhận.

Ngoài các thư viện đã sử dụng sẵn trong `defs.h`, sinh viên KHÔNG được sử dụng thêm bất kỳ thư viện nào khác. Khi nộp bài, sinh viên chỉ nộp đúng một file *sword.cpp* (tên file phải được viết thường). **Tất cả các file nộp khác file *sword.cpp* sẽ bị tự động xóa khi chấm bài.** File được nộp phải là file chương trình gốc, sinh viên không được nén file khi nộp bài. Sinh viên phải kiểm tra chương trình của mình trên Cygwin trước khi nộp.

7. Thay đổi so với phiên bản trước

Loại bỏ điều kiện giao tranh của hiệp sĩ khi không có gươm ở mục d. Đồng nghĩa với việc sự kiện nào không đề cập đến chuyện dùng kiếm thì hiệp sĩ sẽ đấu như bình thường theo mô tả ở mục d.

8. Xử lý 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.

- 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.

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.