

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 1

**CÔNG CHÚA GUINEVERE**

*Phiên bản 1.0*

---

TP. HỒ CHÍ MINH, THÁNG 02/2023

# 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 thực:

- Các cấu trúc rẽ nhánh
- Các cấu trúc lặp
- Mảng 1 chiều và mảng 2 chiều
- Xử lý chuỗi ký tự
- Hàm và lời gọi hàm
- Các thao tác đọc/ghi tập tin

## 2 Dẫn nhập

Công chúa Guinevere, con gái của vua Leodegrance, là một giai nhân tuyệt sắc, vợ hứa hôn của vua Arthur thần thánh. Lời đồn đại về nhan sắc của nàng đã vượt qua biên giới của nước Anh, đến tai của con rồng Bowser hung dữ ngự trị ở vương quốc Koopa xa xôi. Bowser bèn bắt cóc Guinevere mang về Koopa để buộc nàng làm vợ.

Vào thời điểm này, Arthur chỉ vừa đánh bại Cerdic và bước lên ngai vàng của nước Anh. Hay tin người yêu bị bắt cóc, vua Arthur thoát tiên muốn đi giải cứu Guinevere. Thế nhưng rời bỏ nước Anh vào lúc này đồng nghĩa với một lần nữa vương quốc rơi vào tình trạng loạn lạc và cát cứ, sự ổn định và yên bình mà Arthur rất khó khăn mới tạo dựng được rất có thể sẽ lại tiêu tan.

Các hiệp sĩ Bàn Tròn, dẫn đầu là hiệp sĩ Lancelot, đề nghị vua Arthur tiếp tục ở lại ổn định nội tình nước Anh. Việc giải cứu công chúa Guinevere sẽ do hiệp sĩ Bàn Tròn đảm nhiệm. Liệu vua Arthur có từ bỏ vương quốc vì tình yêu, liệu các hiệp sĩ Bàn Tròn có thể chiến thắng được con rồng Bowser ghê gớm để giải cứu công chúa Guinevere, tất cả sẽ được giải đáp trong bài tập lớn này.

Trong bài tập lớn này, sinh viên sẽ được cung cấp một file chứa dữ liệu nhập, bao gồm thông tin về hiệp sĩ Bàn Tròn giải cứu Guinevere và các dữ liệu về hành trình đến vương quốc Koopa. Chương trình sẽ cập nhật các chỉ số của Hiệp sĩ trong quá trình giải cứu Guinevere (sẽ



được trình bày rõ hơn ở phần 4).

### 3 Dữ liệu nhập

Dữ liệu nhập của chương trình được chứa trong một file, tên file sẽ được lưu trong biến `file_input`. File này sẽ chứa các thông tin như sau:

- Thông tin về hiệp sĩ Bàn tròn, ví dụ như sau:

172\_2\_0\_1\_1

- Hành trình đến Koopa, ví dụ như sau:

```
1_2_9_18_99
```

- Các tên file dữ liệu thêm, ví dụ như sau:

```
mush_ghost,,merlin_pack
```

Như vậy file nhập sẽ bao gồm 3 dòng. Dòng đầu tiên sẽ mô tả thông tin của hiệp sĩ tham gia giải cứu Guinevere, có định dạng như sau:

```
HP_level_remedy_maidenkiss_pheonixdown
```

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.
- **remedy**: Số thuốc phục hồi Remedy mà hiệp sĩ mang theo, là một số nguyên có giá trị từ 0 đến 99.
- **maidenkiss**: Số thuốc MaidenKiss mà hiệp sĩ mang theo, là một số nguyên có giá trị từ 0 đến 99.
- **phoenixdown**: Số giọt nước mắt phượng hoàng PhoenixDown mà hiệp sĩ có được, là một số nguyên có giá trị từ 0 đến 99.

Dòng thứ hai của file nhập sẽ mô tả hành trình đến Koopa. Hành trình đến Koopa sẽ bao gồm nhiều sự kiện, mỗi sự kiện sẽ 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 Bảng 1. Số sự kiện là không cố định, có thể thay đổi tùy theo test case. Một sự kiện có thể xảy ra nhiều lần.

Dòng thứ ba của file nhập sẽ gồm tên các file cần dùng trong một số sự kiện, dòng thứ 3 có định dạng như sau:

```
<file_mush_ghost>,<file_asclepius_pack>,<file_merlin_pack>
```

Trong đó:

- **<file\_mush\_ghost>** là tên file sử dụng trong sự kiện 13<ms>.
- **<file\_asclepius\_pack>** là tên file sử dụng trong sự kiện 19.

- `<file_merlin_pack>` là tên file sử dụng trong sự kiện 18.

Chi tiết của từng sự kiện được nêu trong Mục 4. 3 tên file được phân cách bởi 1 dấu phẩy. Nếu hành trình đến Koopa không có sự kiện cần sử dụng file tương ứng, tên file có thể là chuỗi rỗng, hiệp sĩ không cần quan tâm đến tên file này.

### Ví dụ 3.1

Với dữ liệu nhập là

```
172_2_0_1_1  
1_2_2_18_99
```

Hiệp sĩ Bàn Tròn tham gia giải cứu công chúa sẽ có **HP** là 172, level là 2, hiệp sĩ không có Remedy, có 1 MaidenKiss và 1 PhoenixDown. Trên đường đến Koopa, hiệp sĩ sẽ lần lượt gặp sự kiện như sau:

- Sự kiện 1: gặp gấu MadBear.
- Sự kiện 2: gặp cướp Bandit.
- Sự kiện 3: gặp thêm một cướp Bandit.
- Sự kiện 4: gặp phù thủy Merlin.
- Sự kiện 5: gặp rồng Bowser.

Đồng thời, **MaxHP** của hiệp sĩ cũng là 172 (nghĩa là **HP** của hiệp sĩ sẽ không bao giờ vượt quá 172 trong suốt cuộc hành trình).

## 4 Mô tả

Trong quá trình xử lý các sự kiện, hàm phải cập nhật giá trị phù hợp cho 5 chỉ số **HP**, **level**, **remedy**, **maidenkiss**, **phoenixdown**. Chỉ số **rescue** thể hiện hiệp sĩ có giải cứu thành công công chúa Guinevere hay không. **rescue** cần được cập nhật giá trị phù hợp sau khi xử lý các sự kiện: **rescue** = -1 nếu hành trình chưa kết thúc, **rescue** = 1 nếu hành trình kết thúc và giải cứu công chúa thành công, **rescue** = 0 nếu không giải cứu được công chúa (ví dụ: **HP** của hiệp sĩ nhỏ hơn 0 sau khi chiến đấu với đối thủ).

Bảng 1: Các sự kiện trên hành trình đến Koopa

Mã sự kiện	Ý nghĩa
0	Bowser đầu hàng và trả lại công chúa
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 phù thủy Shaman
7	Gặp Siren Vajsh
11	Nhật được nắm tăng lực MushMario
12	Nhật được nắm Fibonacci MushFib
13<ms>	Nhật được nắm ma MushGhost
15	Nhật được thuốc phục hồi Remedy
16	Nhật được thuốc giải MaidenKiss
17	Nhật được giọt nước mắt phượng hoàng PhoenixDown
18	Gặp phù thủy Merlin
19	Gặp thần Asclepius
99	Gặp Bowser

## Ví dụ 4.1

Với dữ liệu nhập là:

```
172_2_0_1_0
0
```

Hiệp sĩ Bàn Tròn chỉ gặp một sự kiện là rồng Bowser đầu hàng và trả lại công chúa. Các chỉ số giữ nguyên và có giá trị như sau:

- *HP* = 172
- *level* = 2
- *remedy* = 0
- *maidenkiss* = 1
- *phoenixdown* = 0
- *rescue* = 1

Tùy theo các sự kiện diễn ra trên đường đi, các chỉ số *HP*, *level*, *remedy*, *maidenkiss* và *phoenixdown* của hiệp sĩ sẽ có sự thay đổi, cụ thể được mô tả như sau:

1. Nếu gặp sự kiện có mã là 0, hiệp sĩ giải cứu công chúa thành công, hành trình của hiệp sĩ chấm dứt ngay lập tức, mặc dù vẫn còn có các sự kiện khác xảy ra tiếp sau sự kiện này trong dữ liệu nhập.

#### Ví dụ 4.2

Với dữ liệu nhập là

```
172_2_0_1_0
0_10_5_15_21_99
```

Các chỉ số có giá trị là:

- $HP = 172$
- $level = 2$
- $remedy = 0$
- $maidenkiss = 1$
- $phoenixdown = 0$
- $rescue = 1$

2. 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  $i$  (sự kiện đầu tiên tương ứng với  $i = 1$ ),  $levelO$  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  $levelO$  của đối thủ, hiệp sĩ chiến thắng. Khi chiến thắng  $level$  của hiệp sĩ sẽ được tăng lên 1 đơn vị - tuy nhiên nếu  $level$  của hiệp sĩ đã là 10 thì  $level$  không thể tăng thêm. Nếu  $level$  của hiệp sĩ bằng  $levelO$ , trận đấu hoà, hiệp sĩ tiếp tục cuộc hành trình nhưng chỉ số  $level$  không được tăng thêm.

Nếu  $level$  của hiệp sĩ nhỏ hơn  $levelO$  của đối thủ,  $HP$  của hiệp sĩ sẽ được tính lại như sau:

$$damage = baseDamage * levelO * 10 \quad (3)$$

$$HP = HP - damage \quad (4)$$

Trong đó, baseDamage sẽ tùy thuộc vào đối thủ, được mô tả ở Bảng 2.

Lưu ý rằng **HP** sẽ luôn là số nguyên khi tính bằng Công thức 3 và 4. Nếu **HP** nhỏ hơn hoặc bằng 0 sau khi tính bằng Công thức 3 và hiệp sĩ không có nước mắt phượng hoàng PhoenixDown, hiệp sĩ sẽ không thể đi tiếp các sự kiện sau đó nếu có, chương trình sẽ kết thúc và hiệp sĩ không thể cứu được công chúa. Trong trường hợp hiệp sĩ có PhoenixDown, bảo bối này sẽ được tự động sử dụng (chỉ số **phoenixdown** sẽ được giảm đi 1), **HP** của hiệp sĩ sẽ được phục hồi về giá trị **MaxHP** ban đầu và hiệp sĩ tiếp tục hành trình.

Bảng 2: Chỉ số baseDamage của các đối thủ

Đối thủ	baseDamage
MadBear	1
Bandit	1.5
LordLupin	4.5
Elf	7.5
Troll	9.5

#### Ví dụ 4.3

Với dữ liệu nhập là:

```
172_1_0_1_0
5_2
```

Ở sự kiện 1, hiệp sĩ gặp quỷ khổng lồ Troll có **levelO** là 1. Do **level** của hiệp sĩ lúc này cũng đang là 1, trận đấu hoà và hiệp sĩ chuyển sang sự kiện tiếp theo. Ở sự kiện 2, hiệp sĩ gặp Bandit có **levelO** là 7. Do **level** của hiệp sĩ chỉ là 1, hiệp sĩ thua, **HP** của hiệp sĩ sẽ còn lại là  $172 - 1.5 * 2 * 10 = 142$ . Đến đây do không còn sự kiện nào xảy ra tiếp theo, hiệp sĩ đến được Koopa và giải cứu công chúa thành công, các chỉ số có giá trị là:

- **HP** = 142
- **level** = 1
- **remedy** = 0
- **maidenkiss** = 1
- **phoenixdown** = 0
- **rescue** = 1



#### Ví dụ 4.4

Với dữ liệu nhập là:

```
152_1_0_1_0  
3_5_7
```

Sau khi đánh hoà với LordLupin ở sự kiện 1, ở sự kiện 2 hiệp sĩ đánh với Troll có *level* là 2 và chịu thua do có *level* thấp hơn, *HP* của hiệp sĩ do đó sẽ trở thành  $152 \sim 8.5 * 2 * 10 = -18 < 0$ . Do đó, chương trình kết thúc và hiệp sĩ không giải cứu được công chúa. Các chỉ số có giá trị là:

- *HP* = -18
- *level* = 1
- *remedy* = 0
- *maidenkiss* = 1
- *phoenixdown* = 0
- *rescue* = 0

**Ví dụ 4.5**

Với dữ liệu nhập là:

```
152 1 0 1 1  
3 5
```

Tương tự ví dụ trước, sau khi đánh hoà với LordLupin ở sự kiện 1, ở sự kiện 2 hiệp sĩ đánh với Troll có levelO là 2 và chịu thua do có *level* thấp hơn, *HP* của hiệp sĩ do đó sẽ trở thành  $152 \sim 8.5 * 2 * 10 = -18 < 0$ . Do lúc này hiệp sĩ có 1 PhoenixDown, bảo bối này sẽ được tự động sử dụng, *HP* của hiệp sĩ phục hồi về giá trị *MaxHP* ban đầu (152). Đến đây do không còn sự kiện nào tiếp theo, hiệp sĩ đến được Koopa và cứu được công chúa. Các chỉ số có giá trị là:

- *HP* = 152
- *level* = 1
- *remedy* = 0
- *maidenkiss* = 1
- *phoenixdown* = 0
- *rescue* = 1

3. **Nếu gặp phù thủy Shaman (mã sự kiện là 6)**, hiệp sĩ sẽ giao tranh với Shaman. Cách giao tranh tương tự như mô tả trong Mục 2. Nếu hiệp sĩ thắng, level của hiệp sĩ sẽ tăng lên 2 đơn vị, nhưng không được tăng quá 10. Nếu hoà, hiệp sĩ tiếp tục đi tiếp. Nếu thua, hiệp sĩ sẽ bị biến thành tí hon trong 3 sự kiện tiếp theo. Ở trạng thái tí hon, *HP* của hiệp sĩ sẽ bị giảm bằng  $1/5$  *HP* hiện tại (chỉ tính số nguyên). Tuy nhiên, nếu *HP* ban đầu của hiệp sĩ nhỏ hơn 5, *HP* của hiệp sĩ sẽ được tính thành 1. Sau khi hết trạng thái tí hon, *HP* của hiệp sĩ sẽ được nhân 5 trở lại (nếu lúc này *HP* lớn hơn *MaxHP*, *HP* sẽ tự động giảm về *MaxHP*).

Ngay khi bị biến thành tí hon, nếu hiệp sĩ có thuốc Remedy (*remedy*  $\geq 1$ ), hiệp sĩ sẽ tự động dùng thuốc này và trở lại bình thường, *HP* không bị thay đổi, khi đó chỉ số *remedy* của hiệp sĩ bị giảm đi 1. Nếu *HP* của hiệp sĩ giảm xuống bằng hoặc dưới 0 khi hiệp sĩ ở trạng thái tí hon và hiệp sĩ có PhoenixDown để sử dụng, hiệp sĩ sẽ được giải thoát khỏi trạng thái tí hon, *HP* cũng hiệp sĩ sẽ được phục hồi về *MaxHP*.

## Ví dụ 4.6

Với dữ liệu nhập là:

152 1 0 0 0

4 6 5

Sau khi đánh hoà với Elf ở sự kiện 1, ở sự kiện 2 hiệp sĩ đánh với Shaman có levelO là 2 và bị biến thành tí hon do có level thấp hơn, **HP** của hiệp sĩ lúc đó sẽ bị giảm xuống còn 30. Ở sự kiện 3 hiệp sĩ đánh với Troll có levelO là 3 và thua. **HP** của hiệp sĩ khi đó bị giảm còn  $30 - 8.5 * 3 * 10 = -225 < 0$ . Do hiệp sĩ không có PhoenixDown, chương trình kết thúc và hiệp sĩ không giải cứu được công chúa. Các chỉ số có giá trị là:

- **HP** = -225
- **level** = 1
- **remedy** = 0
- **maidenkiss** = 0
- **phoenixdown** = 0
- **rescue** = 0

#### Ví dụ 4.7

Với dữ liệu nhập là:

998\_1\_0\_0\_0

4\_6\_1\_1\_1

Sau khi đánh hoà với Elf ở sự kiện 1, ở sự kiện 2 hiệp sĩ đánh với Shaman có levelO là 2 và bị biến thành tí hon do có level thấp hơn, **HP** của hiệp sĩ lúc đó sẽ bị giảm xuống còn 199. Ở các sự kiện 3, 4, 5 hiệp sĩ liên tục đánh với các MadBear có levelO lần lượt là 3, 4, 5 và đều thua. **HP** của hiệp sĩ khi đó bị giảm còn  $199 - 1 \cdot (3 + 4 + 5) \cdot 10 = 79$ . Đến đây do đã đủ 3 sự kiện, **HP** của hiệp sĩ sẽ tự động phục hồi về  $79 \cdot 5 = 395$ . Do không còn sự kiện nào tiếp theo, hiệp sĩ đến được Koopa và cứu công chúa thành công. Các chỉ số có giá trị là:

- **HP** = 395
- **level** = 1
- **remedy** = 0
- **maidenkiss** = 0
- **phoenixdown** = 0
- **rescue** = 1

## Ví dụ 4.8

Với dữ liệu nhập là:

998\_1\_2\_0\_0

4\_6\_1

Sau khi đánh hoà với Elf ở sự kiện 1, ở sự kiện 2 hiệp sĩ đánh với Shaman có levelO là 2 và bị biến thành tí hon do có level thấp hơn, **HP** của hiệp sĩ lúc đó sẽ bị giảm xuống còn 199. Do hiệp sĩ có 2 remedy, một viên thuốc giải sẽ được tự động sử dụng, khi đó **HP** của hiệp sĩ sẽ là  $199 * 5 = 995$ . Ở sự kiện 3, hiệp sĩ đánh thua MadBear có levelO là 3 và **HP** của hiệp sĩ còn  $995 - 1 * 3 * 10 = 965$ . Các chỉ số có giá trị là:

- **HP** = 965
- **level** = 1
- **remedy** = 1
- **maidenkiss** = 0
- **phoenixdown** = 0
- **rescue** = 1

4. **Nếu gặp Siren Vajsh (mã sự kiện là 7)**, hiệp sĩ giao tranh với Vajsh. Cách giao tranh tương tự như mô tả trong Mục 2. Nếu hiệp sĩ thắng, level của hiệp sĩ sẽ tăng 2 đơn vị nhưng không được tăng quá 10, sau đó sẽ đi tiếp. Nếu hòa, hiệp sĩ tiếp tục đi tiếp. Nếu thua, hiệp sĩ sẽ bị biến thành ếch trong 3 sự kiện tiếp theo. Ngay khi bị biến thành ếch, nếu hiệp sĩ có thuốc MaidenKiss, hiệp sĩ sẽ tự động dùng thuốc này và trở về bình thường, level không bị thay đổi, chỉ số maidenkiss sẽ bị giảm đi 1. Khi biến thành ếch, level của hiệp sĩ sẽ bị giảm xuống thành 1. Sau khi biến thành người trở lại, level của hiệp sĩ được phục hồi về giá trị cũ trước khi bị biến thành ếch (thậm chí lúc bị biến thành ếch level của hiệp sĩ có tăng lên thì khi biến thành người level của hiệp sĩ cũng sẽ trở về giá trị cũ trước khi bị biến thành ếch).

Khi đang bị biến thành tí hon hoặc thành ếch, nếu hiệp sĩ nhặt được thuốc giải tương ứng (Remedy hoặc MaidenKiss), thuốc giải sẽ được tự động sử dụng ngay. Shaman và Vajsh sẽ bỏ qua không giao tranh với hiệp sĩ nếu hiệp sĩ đang bị biến thành tí hon hoặc ếch.

## Ví dụ 4.9

Với dữ liệu nhập là:

```
998_1_0_0_0  
4_6_7
```

Sau khi đánh hoà với Elf ở sự kiện 1, ở sự kiện 2 hiệp sĩ đánh với Shaman có levelO là 2 và bị biến thành tí hon do có level thấp hơn, **HP** của hiệp sĩ lúc đó sẽ bị giảm xuống còn 199. Ở sự kiện 3, hiệp sĩ gặp Vajsh, do hiệp sĩ đang bị biến thành tí hon nên Vajsh sẽ bỏ qua không giao tranh với hiệp sĩ. Các chỉ số có giá trị là:

- **HP** = 199
- **level** = 1
- **remedy** = 0
- **maidenkiss** = 0
- **phoenixdown** = 0
- **rescue** = 1

5. Nếu hiệp sĩ nhặt được nấm tăng lực MushMario (mã sự kiện là 11), **HP** của hiệp sĩ sẽ tăng lên như sau:

$$HP = HP + (s1\%100)$$

Trong đó:  $n1 = ((level + phoenixdown)\%5 + 1) * 3$ ,  $s1$  là tổng của  $n1$  số lớn nhất trong tập hợp các số nguyên dương lẻ có 2 chữ số. Ví dụ: nếu  $n1 = 3$  thì  $s1 = 99 + 97 + 95 = 291$ . Sau đó, **HP** được tăng lên số nguyên tố gần nhất. Sau khi tăng lên, nếu **HP** lớn hơn **MaxHP**, **HP** sẽ được tự động giảm xuống **MaxHP**.

6. Nếu hiệp sĩ nhặt được nấm Fibonacci MushFibo (mã sự kiện là 12), **HP** của hiệp sĩ sẽ giảm xuống số Fibonacci gần nhất nếu **HP** > 1. Nếu **HP** = 1 thì các chỉ số không bị thay đổi. Số Fibonacci là một số nằm trong dãy Fibonacci. Đây là dãy vô hạn các số tự nhiên bắt đầu bằng 1 và 1, sau đó các số tiếp theo sẽ bằng tổng của 2 số liền trước nó.
7. Nếu hiệp sĩ nhặt được nấm ma MushGhost (mã sự kiện bắt đầu là 13, sau đó là chuỗi <ms>), hiệp sĩ có thể gặp một hoặc nhiều loại nấm ma khác nhau dựa theo chuỗi <ms>. Ví dụ:
- Nếu mã sự kiện là 131, <ms> là 1, hiệp sĩ gặp nấm ma Loại 1.
  - Nếu mã sự kiện là 1342, <ms> là 42, hiệp sĩ gặp nấm ma Loại 4 rồi gặp nấm ma Loại

2. Ngay sau khi gặp mỗi loại nắm ma, **HP** của hiệp sĩ cần phải cập nhật cho phù hợp trước khi gặp nắm ma tiếp theo. Nếu **HP** vượt quá **MaxHP** thì cũng cần cài đặt lại bằng **MaxHP** trước khi gặp nắm ma tiếp theo. Nếu **HP** xuống dưới 0; nếu thuốc **phoenixdown** > 0 thì thuốc sẽ được tự động sử dụng, **HP** được phục hồi về **MaxHP**, hiệp sĩ sẽ gặp nắm ma tiếp theo; nếu không còn thuốc **phoenixdown**, chương trình sẽ dừng lại, hiệp sĩ không gặp nắm ma tiếp theo.

Khi gặp nắm ma, cần thao tác với một dãy số được lưu trong file có tên **<file\_mush\_ghost>**. Tên file nằm ở dòng 3 của file đầu vào (Xem mô tả Mục 3). File có định dạng gồm 2 dòng: dòng 1 chứa 1 số nguyên dương  $n2$ , dòng 2 chứa  $n2$  số nguyên cách nhau bởi 1 dấu phẩy,  $n2 \leq 100$ . Có 4 loại nắm ma tương ứng với 4 mã số 1, 2, 3, 4. **HP** của hiệp sĩ sẽ thay đổi ứng với từng loại nắm như sau:

- **Nắm ma loại 1 (mã số 1):** Gọi *maxi* và *mini* lần lượt là vị trí cuối cùng (vị trí tính từ 0) của số lớn nhất và số nhỏ nhất trong  $n2$  số nguyên ở dòng 2.

$$\mathbf{HP} = \mathbf{HP} - (maxi + mini)$$

- **Nắm ma loại 2 (mã số 2):** Kiểm tra xem dãy số có hình dạng núi hay không. Một dãy số có hình dạng núi là dãy gồm các số tăng dần đến số lớn nhất (gọi số này là đỉnh núi), sau đó giảm dần. Tính chất tăng dần và giảm dần yêu cầu là tăng nghiêm ngặt ( $a < b$ ) và giảm nghiêm ngặt ( $a > b$ ) với  $a, b$  là 2 giá trị liên tiếp trong mảng. Hình dạng núi chấp nhận đỉnh núi ở phần tử đầu tiên hoặc phần tử cuối cùng. Gọi *mtx* và *mti* lần lượt là giá trị và vị trí của đỉnh núi. Nếu dãy số không có hình dạng núi thì  $mtx = -2$ ,  $mti = -3$ .

$$\mathbf{HP} = \mathbf{HP} - (mtx + mti)$$

- **Nắm ma loại 3 (mã số 3):** Với mỗi số nguyên  $xi$  trong dãy số ở dòng 2 của file, thực hiện phép biến đổi:

- $xi = -xi$  nếu  $xi < 0$
- $xi = (17 * xi + 9) \% 257$

Trong dãy số sau khi biến đổi: gọi *maxi2* và *mini2* lần lượt là vị trí đầu tiên (tính từ 0) của số lớn nhất và số nhỏ nhất.

$$\mathbf{HP} = \mathbf{HP} - (maxi2 + mini2)$$

- **Nắm ma loại 4 (mã số 4):** Thực hiện biến đổi dãy số như nắm ma loại 3. Gọi *max2\_3x* và *max2\_3i* lần lượt là số lớn thứ hai và vị trí đầu tiên (tính từ 0)

của số lớn thứ 2 trong 3 số đầu tiên của dãy, nếu không tồn tại số lớn thứ hai thì  $\text{max2\_}3x = -5, \text{max2\_}3i = -7$ .

$$HP = HP - (\text{max2\_}3x + \text{max2\_}3i)$$

8. Nếu hiệp sĩ nhặt được **Remedy**, **MaidenKiss** hoặc **PhoenixDown** (mã sự kiện tương ứng là **15**, **16** và **17**), các chỉ số **remedy**, **maidenkiss** và **phoenixdown** tương ứng sẽ được tăng lên 1. Tuy nhiên các chỉ số này không được tăng quá giá trị 99.
9. Nếu hiệp sĩ gặp **Asclepius** (mã sự kiện là **19**, **Asclepius** là vị thần của thuốc, ông sẽ tặng hiệp sĩ một túi đồ gồm các loại thuốc khác nhau như **remedy**, **maidenkiss**, **phoenixdown**. Thông tin của túi thuốc được lưu trong một file, tên file được lưu trong biến **file\_ asclepius\_ pack**, định dạng nội dung file như sau:

- Dòng 1 chứa 1 số nguyên dương  $r1$ .
- Dòng 2 chứa 1 số nguyên dương  $c1$ .
- $r1$  dòng tiếp theo, mỗi dòng  $c1$  số nguyên cách nhau bởi 1 khoảng trắng. Mỗi số nguyên biểu diễn một loại thuốc. **Asclepius** chỉ cho hiệp sĩ lấy thuốc có số nguyên biểu diễn là 16, 17, 18 tương ứng với **remedy**, **maidenkiss**, **phoenixdown**. Đồng thời, **Asclepius** chỉ cho Hiệp sĩ lấy tối đa 3 thuốc trên một dòng, cách lấy thuốc là đi từ đầu đến cuối mỗi dòng.

Trong hành trình đến Koopa, hiệp sĩ chỉ được **Asclepius** cho túi thuốc tối đa một lần và ở lần đầu tiên gặp. Nếu hiệp sĩ gặp **Asclepius** ở các lần sau, hiệp sĩ sẽ bỏ qua và không làm gì cả. Sau khi nhận được túi thuốc, nếu hiệp sĩ đang ở trong trạng thái tí hon hoặc éch, và có thuốc tương ứng, thì thuốc tương ứng sẽ được tự động sử dụng.

10. Nếu **HP** ban đầu của hiệp sĩ là **999**, hiệp sĩ đó chính là vua **Arthur** từ bỏ ngai vàng nước Anh để đi cứu **Guinevere**. **Arthur** sẽ đánh thắng tất cả mọi đối thủ trong tất cả các cuộc giao tranh.
11. Nếu **HP** ban đầu của hiệp sĩ là số nguyên tố, hiệp sĩ này là một **Lancelot**. **Lancelot** đánh thắng tất cả **MadBear**, **Bandit**, **LordLupin**, **Elf**, **Troll**, **Shaman** và **Vajsh** bất chấp levelO của đối thủ.
12. Hiệp sĩ có thể gặp chính **Bowser** trong cuộc hành trình (mã sự kiện **99**). **Bowser** chỉ có thể đánh bại bởi **Arthur**, **Lancelot** có level lớn hơn hoặc bằng 8, hoặc hiệp sĩ **Bàn Tròn** có level là 10. Khi đánh bại được **Bowser**, level của hiệp sĩ sẽ được tăng lên thành 10. Nếu thua **Bowser**, chương trình chấm dứt và hiệp sĩ không giải cứu được công chúa. Lưu ý là hiệp sĩ vẫn tiếp tục cuộc hành trình ngay cả khi đánh bại **Bowser**.
13. Nếu hiệp sĩ gặp phù thủy **Merlin** (mã sự kiện là **18**), **Merlin** sẽ cho hiệp sĩ một túi đồ. Trong túi đồ này có một số món đồ được **Merlin** phù phép có khả năng tăng **HP**



cho hiệp sĩ. Thông tin của túi đồ được lưu trong một file, tên file được lưu trong biến ***file\_merlin\_pack***, định dạng nội dung file như sau:

- Dòng 1 chứa 1 số nguyên dương  $n9$ .
- $n9$  dòng tiếp theo, mỗi dòng chứa một chuỗi biểu diễn tên của món đồ. Một món đồ được phù phép là món đồ có tên chứa đủ 6 chữ cái trong tên của Merlin, 6 chữ cái này không cần quan tâm viết hoa hay viết thường.

Với mỗi món đồ được phù phép, hiệp sĩ được tăng 2 **HP** nhưng không được tăng quá **MaxHP**. Tuy nhiên, nếu tên món đồ nào có chứa chuỗi "Merlin" hoặc "merlin" thì hiệp sĩ được tăng 3 **HP** nhưng không được tăng quá **MaxHP**.

Merlin chỉ cho hiệp sĩ túi đồ trong lần đầu tiên gặp.

## 5 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, knight.h, knight.cpp, và các file dữ liệu đọc mẫu. Sinh viên sẽ chỉ nộp 2 tập tin là knight.h và knight.cpp nên không được sửa đổi tập tin main.h khi chạy thử chương trình.
3. Sinh viên sử dụng câu lệnh sau để biên dịch:

**g++ -o main main.cpp knight.cpp -I . -std=c++11**

Sinh viên sử dụng câu lệnh sau để chạy chương trình:

**./main tc1 \_input**

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 knight.cpp, thêm option -std=c++11, -I .

4. 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.

5. Sửa đổi các file `knight.h`, `knight.cpp` để hoàn thành bài tập lớn này và đảm bảo hai yêu cầu sau:

- Hiện thực hàm **`adventureToKoopa`** mô tả hành trình hiệp sĩ đi đến Koopa và gặp các sự kiện khác nhau. Cách thức xử lý sự kiện như mô tả đề bài. Trong hàm này, tham số **`file_input`** chứa tên của file nhập đầu vào cho chương trình, các tham số còn lại có ý nghĩa giống như các chỉ số của hiệp sĩ và chỉ số **`rescue`**. Trong hàm **`adventureToKoopa`**, sau mỗi sự kiện, chương trình cần in ra các chỉ số hiện tại bằng cách gọi hàm `display` được cung cấp sẵn.
- Chỉ có 1 lệnh **`include`** trong tập tin `knight.h` là **`#include "main.h"`** và một include trong tập tin `ConcatStringList.cpp` là **`#include "knight.h"`**. Ngoài ra, không cho phép có một **`#include`** nào khác trong các tập tin này.

6. Sinh viên được khuyến khích viết thêm các hàm để hoàn thành BTL này.

## 6 Nộp bài

Sinh viên chỉ nộp 2 tập tin: `knight.h` và `knight.cpp`, trước thời hạn được đưa ra trong đường dẫn "Assignment 1 - 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.

## 7 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.