

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

**SHERLOCK HOLMES
THE AWAKENED - Phần 1**

TP. HỒ CHÍ MINH, THÁNG 11/2023

ĐẶC TẢ BÀI TẬP LỚN

Phiên bản 1.1

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

Bài tập lớn này được lấy cảm hứng từ trò chơi Sherlock Holmes: The Awakened, được phát triển bởi Frogwares

Ngày 6 tháng 9 năm 1894, Sherlock Holmes chán nản nhìn ra cửa sổ. Vị thám tử đang trải qua một giai đoạn khó khăn, khi mà không có một vụ án nào xứng đáng với tài năng của mình. Ông chuẩn bị đạt được điều mình muốn.

3 Nhiệm vụ

Sinh viên được yêu cầu xây dựng một chương trình giả tưởng trên ngôn ngữ C++ để mô phỏng lại quá trình giải quyết các vụ án của Holmes và Watson: The Awakened, thông qua các nhiệm vụ được mô tả bên dưới. Mỗi nhiệm vụ được yêu cầu viết hàm tương ứng, các tham số của cho hàm này sẽ được cho trong mô tả của yêu cầu nhiệm vụ.

3.1 Nhiệm vụ 1: Những điều bất thường ở bến cảng

Tin tức nhanh chóng đến tai Holmes về sự biến mất của một người hầu Maori trẻ tuổi làm việc cho Thuyền trưởng Stenwick giàu có. Theo đuổi các manh mối dẫn họ đến các bến cảng gần

sông Thames, nơi Holmes và John Watson sớm biết rằng nhiều "vụ mất tích" tương tự đã xảy ra.

Khi đến bến tàu, Holmes đã kiểm tra các tài liệu của bến tàu, đó là những ghi chép về các tàu đến và rời bến tàu. Tuy nhiên, vụ nổ của Tàu Đỏ đã khiến các hồ sơ bị đốt cháy và trở nên lộn xộn. Holmes và Watson đã lục lọi qua những mảnh giấy bị cháy và phát hiện ra những tài liệu giả mạo bất thường của bến tàu.

3.1.1 Thời điểm không hợp lệ

Sinh viên được yêu cầu viết một hàm để tìm các thời điểm không hợp lệ trong tài liệu của bến tàu. Mô tả của hàm như sau:

- Tên hàm: **invalidTime**.
- Tham số đầu vào:
 - **schedule**: Chuỗi chứa tên tập tin, tập tin này chứa thông tin về lịch tàu tại bến.
- Cấu trúc của tập tin **schedule**:
 - Dòng 1: một chuỗi đã định dạng **Number of ships: <n>**, trong đó <n> gồm 3 ký tự số. <n> sẽ đại diện cho một số nguyên trong khoảng từ [1, 999]. Ví dụ, $n1 = "001"$ đại diện cho số nguyên 1; $n1 = "010"$ đại diện cho số nguyên 10; $n1 = "123"$ đại diện cho số nguyên 123.
 - Từ dòng 2 trở đi, mỗi dòng sẽ được định dạng theo cú pháp sau:

ship_name, DD/MM hh:mm - DD/MM hh:mm

Trong đó:

- * ship_name: Tên của tàu, biểu diễn bằng chữ in hoa, chữ thường và số (không có khoảng trắng).
- * DD: Biểu thị ngày trong tháng, biểu diễn bằng 2 chữ số, với khoảng giá trị từ 01 đến 31. Số ngày trong tháng phụ thuộc vào tháng hiện tại.
- * MM: Biểu thị tháng, biểu diễn bằng 2 chữ số, với khoảng giá trị từ 01 đến 12.
- * hh: Biểu thị giờ trong ngày, biểu diễn bằng 2 chữ số, với khoảng giá trị từ 00 đến 23.
- * mm: Biểu thị phút, biểu diễn bằng 2 chữ số, với khoảng giá trị từ 00 đến 59.
- * Một tổ hợp DD/MM hh:mm được coi là một thời điểm. Thời điểm đầu tiên là thời điểm tàu cập bến, và thời điểm thứ hai là thời điểm tàu rời bến.

Do các mảnh giấy bị cháy, một số chữ số trong tài liệu bị hỏng và không thể đọc được. Các chữ số không đọc được này được ký hiệu bằng ký tự '?'.

- **Yêu cầu hàm:** Hàm đọc dữ liệu từ tập tin **schedule**. Nếu $\langle n \rangle$ là một chuỗi không hợp lệ, hàm không thực hiện gì và trả về giá trị -1. Nếu chuỗi $\langle n \rangle$ là hợp lệ, ta đếm số lượng tàu có thời điểm đến và rời bến hợp lệ. Nếu trong lịch trình có ký tự '?', và có một cách thay thế tất cả các ký tự '?' bằng các chữ số sao cho thời điểm đến và rời bến trở nên hợp lệ, thì con tàu đó được coi là hợp lệ. Trong hàm này, sinh viên không cần phải kiểm tra xem khoảng thời gian giữa lúc đến và lúc rời bến có hợp lệ hay không; điều này sẽ được xem xét trong phần sau. Hàm trả về số lượng tàu hợp lệ trong tập tin **schedule**. Nếu không có tàu nào hợp lệ, hàm trả về -1. Số ngày trong tháng được tính theo năm 1864.

Ví dụ 1: Các ví dụ về $\langle n1 \rangle$ là chuỗi không hợp lệ:

- **Number of ships: 11x:** không hợp lệ do có ký tự cuối cùng là 'x', không phải là một chữ số.
- **Number of ships: 000:** không hợp lệ do khi chuyển qua số nguyên sẽ là số 0, không thuộc khoảng $[1, 999]$.

Trong hai trường hợp này, hàm trả về giá trị -1.

Ví dụ 2: Tập **schedule** có nội dung như sau:

1	Number of ships: 004
2	NO22, 12/02 14:00 - 12/02 18:00
3	NVD34, 14/0? 16:1? - ??/03 ?2:00
4	13Se, 15/13 16:00 - 12/01 14:61
5	ko772, 4?/1? ??:?? - 12/05 3?:25

Theo đó:

- Tàu ở dòng 2: Tất cả thời điểm đều hợp lệ. Do đó, tàu này là hợp lệ.
- Tàu ở dòng 3: Chúng ta có thể thay thế ký tự '?' bằng 14/03 16:15 - 14/03 12:00. Do đó, tàu này là hợp lệ.
- Tàu ở dòng 4: Không hợp lệ vì không có tháng 13, và số phút vượt quá giới hạn 59.
- Tàu ở dòng 5: Không hợp lệ vì không có cách thay thế ký tự '?' sao cho thời điểm đến và rời bến trở nên hợp lệ.

Hàm trả về giá trị: 2.

3.1.2 Thời gian lâu nhất

Sau khi loại bỏ những chuyến tàu không hợp lệ, Holmes tiếp tục quan sát các chuyến tàu đáng ngờ bằng cách xác định chuyến tàu nào đã ở bến lâu nhất.

Sinh viên được yêu cầu viết một hàm để tìm thời gian lưu trú lâu nhất trong tài liệu bến tàu. Mô tả về hàm như sau:

- Tên hàm: **longestDuration**.
- Tham số đầu vào:
 - **schedule**: Chuỗi chứa tên tập tin, tập tin này chứa thông tin về lịch tàu tại bến.
- Cấu trúc của tập tin **schedule**: Tương tự như hàm **invalidTime**.
- Yêu cầu về hàm: Hàm đọc dữ liệu từ tập tin **schedule**. Nếu **<n>** là một chuỗi không hợp lệ, hàm sẽ không làm gì và trả về một chuỗi rỗng (""). Nếu chuỗi **<n>** là hợp lệ, chúng ta tìm chuyến tàu đã ở bến với thời gian lưu trú lâu nhất trong số tất cả các chuyến tàu. Cụ thể:
 - Nếu thời gian đến và rời bến của tàu hợp lệ, thời gian lưu trú của tàu tại bến bằng hiệu giữa thời gian rời bến và thời gian đến bến.
 - Nếu có ký tự '?' trong lịch trình, chúng ta phải tìm cách thay thế những ký tự '?' này để tối đa hóa thời gian tàu ở bến. Nói cách khác, chúng ta cần tìm cách tối đa hóa hiệu giữa hai thời điểm.
 - Nếu thời gian đến bến của tàu lớn hơn thời gian rời bến của tàu, tàu sẽ là không hợp lệ. Nếu không có tàu hợp lệ nào tồn tại, hàm trả về một chuỗi rỗng ("").
 - Nếu có một tàu có thời gian lưu trú lâu nhất tại bến, hàm trả về tên của tàu đó. Nếu có nhiều hơn một tàu có thời gian lưu trú lâu nhất tại bến, hàm trả về tên của tàu có thời gian lưu trú lâu nhất mà nằm cuối cùng trong danh sách.
 - Số ngày trong tháng được tính theo năm 1864.

Ví dụ 3: Với tập tin **schedule** có nội dung như sau:

Dòng	Nội dung	Thời gian lưu trú lâu nhất (tính bằng phút)
1	Number of ships: 007	
2	NO22, 12/02 14:00 - 12/02 18:00	240
3	NVD34, 14/0? 16:1? - ??/03 ?2:00	111230
4	13Se, 15/13 16:00 - 12/01 14:61	Không hợp lệ
5	ko772, 4?/1? ??:?? - 12/05 3?:25	Không hợp lệ
6	tyt8, 11/1? ??:?? - 12/04 1?:25	Không hợp lệ
7	bor08, ??/? ??:?? - 12/04 23:25	148285
8	gate12, ??/? ??:?? - 12/04 23:25	148285

Theo đó:

- Tàu ở dòng 3: Chúng ta có thể thay thế ký tự '?' để đạt được thời gian lưu trú lâu nhất tại bến như sau: 14/01 16:10 - 31/03 22:00. Thời gian tại bến là 111230 phút.
- Tàu ở dòng 6: **Không hợp lệ** vì không có cách thay thế ký tự '?' để làm cho thời điểm rời bến lớn hơn thời điểm đến bến.
- Tàu ở dòng 7 và 8: Chúng ta có thể thay thế ký tự '?' để đạt được thời gian lưu trú lâu nhất như sau: 01/01 00:00 - 12/04 23:25. Thời gian lưu trú là 148285 phút.

Hàm trả về giá trị: gate12.

3.2 Nhiệm vụ 2: Ngôi đền bị bỏ hoang

Cuộc điều tra của Holmes và Watson tại bến cảng dẫn họ đến một mạng lưới các hầm ngầm, mà họ tin rằng đang được sử dụng để vận chuyển người hâu Maori bị bắt cóc và các nạn nhân khác tới ngôi đền ngầm bị bỏ hoang.

Những hầm ngầm này tối tăm và rối ren, và Holmes và Watson phải sử dụng sự quan sát và khả năng suy luận sắc bén của họ để điều hướng qua chúng an toàn. Trên đường, họ gặp phải dấu vết của một nghi lễ kinh dị, bao gồm các vết máu, các biểu tượng hiến tế và quần áo bị bỏ lại.

3.2.1 Khám phá ngôi đền

Sinh viên được giao viết hàm sau đây để mô tả quá trình Holmes và Watson khám phá ngôi đền. Thông tin về hàm như sau:

- Tên hàm: **investigateTemple**.

- Tham số đầu vào:

- **map**: một chuỗi chứa tên tập tin, tập tin này chứa bản đồ của ngôi đền.
- **moveStyle**: Chuỗi chứa hành vi di chuyển của Holmes và Watson. Mỗi ký tự là một trong bốn chữ cái sau:

- * 'U': đi lên
- * 'D': đi xuống
- * 'L': quẹo trái
- * 'R': quẹo phải

Đáng chú ý, Holmes và Watson sẽ di chuyển theo hành vi lặp đi lặp lại. Tức là, nếu Holmes và Watson có một kiểu di chuyển là "LLUU", thì họ sẽ quẹo trái hai lần (LL), sau đó đi lên hai lần (UU), và lặp lại chuỗi này vô tận. Do đó, chuỗi di chuyển thực tế của họ là "LLUULLLUULLLUULLLUULLLU...".

- **stamina**: Một số nguyên thể hiện thể lực của Holmes và Watson.
- **outTime**: Một biến số nguyên được truyền tham khảo, dùng để trả về thời gian mà Holmes và Watson cần để đến bàn thờ.
- **outPath**: Một biến chuỗi được truyền tham khảo, dùng để trả về đường đi mà Holmes và Watson đã đi qua.

- Yêu cầu hàm:

1. Đọc dữ liệu từ tập tin có tên **map** và khởi tạo các giá trị phù hợp.
2. Dựa vào biến **moveStyle**, sinh viên thể hiện các bước di chuyển của Holmes và Watson theo tuần tự qua các ô trên bản đồ. Trong quá trình di chuyển, nếu Holmes và Watson đến một hàng/cột ở biên của bản đồ và bước tiếp theo trong **moveStyle** dẫn họ ra khỏi bản đồ, họ sẽ được chuyển tới đầu bên kia của bản đồ. Ví dụ, nếu Holmes và Watson đang ở vị trí (0,0) và đi lên, vị trí tiếp theo của họ sẽ là (9, 0).
3. Với mỗi bước, thể lực của họ bị giảm đi 1 đơn vị. Nếu Holmes và Watson hết thể lực, họ không thể tiếp tục di chuyển.
4. Nếu Holmes và Watson di chuyển vào cùng một ô quá 3 lần, họ sẽ bị lạc và không thể tiếp tục di chuyển.
5. Holmes và Watson sẽ dừng di chuyển nếu họ tìm thấy bàn thờ, hết thể lực hoặc gặp một số sự kiện cụ thể trong quá trình điều tra.

- Kết quả trả về:

- Hàm trả về **true** nếu Holmes và Watson di chuyển đến bàn thờ. Ngược lại, trả về **false**.

outTime: Số nguyên biểu thị thời gian mà Holmes và Watson cần để đến bàn thờ. Mỗi bước sẽ được tính là một đơn vị thời gian.

outPath: Chuỗi biểu thị đường đi mà Sherlock và Watson đã đi qua. Mỗi điểm trên đường đi được phân tách bởi một ký tự ','. Một đường đi được biểu thị như sau:

$$p_{i1}, p_{j1}; p_{i2}, p_{j2}; \dots; p_{ik}, p_{jk}; \dots; p_{in}, p_{jn}$$

trong đó p_{ik}, p_{jk} là chỉ số của một điểm thứ k theo hàng và theo cột và n là số bước di chuyển mà Sherlock đã đi qua.

3.2.2 Bản đồ của ngôi đền

Bản đồ là một mảng hai chiều kích thước cố định 10x10. Mỗi phần tử của mảng là một ký tự. Mảng hai chiều này biểu thị bản đồ của ngôi đền đang được điều tra.

Bảng dưới đây thể hiện các sự vật/sự kiện trong bản đồ của ngôi đền:

Kí tự	Tên sự vật/sự kiện	Số lượng	Ý nghĩa
S	Holmes và Watson	1	Vị trí bắt đầu của Holmes và Watson
A	Bàn thờ	1	Vị trí của bàn thờ
0	Lối đi	Không giới hạn	Những lối đi trong ngôi đền, có thể đi được
X	Vật cản	Không giới hạn	Những chướng ngại không thể đi qua
R	Gương	Tối đa 1	Đảo ngược hành vi di chuyển
T	Cửa bí mật	Tối đa 2	Tự động di chuyển đến cửa bí mật còn lại
H	Máy thời gian	Tối đa 1	Di chuyển Holmes và Watson đến một thời điểm trong quá khứ
P	Bình thuốc	Tối đa 1	Giảm đi 1 thể lực cho mỗi 2 bước (thay vì 1)
C	Ký hiệu bí ẩn	Tối đa 1	Xoay bản đồ 180 độ
L	Ánh sáng	Tối đa 1	Chiếu sáng vật cản
M	Quái vật	Tối đa 1	Tiêu thụ tâm trí của Holmes và Watson, điều tra thất bại

Tất cả các ký tự được giới thiệu chi tiết như sau:

- Ký tự 'S': Vị trí bắt đầu của Holmes và Watson. Không lưu vị trí này vào outPath ở thời điểm bắt đầu.
- Ký tự 'A': Vị trí bàn thờ. Nếu bạn di chuyển đến điểm này, cuộc điều tra sẽ dừng lại và thành công.
- Ký tự '0': Đây là lối đi trong đền thờ.
- Ký tự 'X': Đây là một chướng ngại vật trong đền thờ. Không thể di chuyển đến những vị trí này. Khi bước vào ô này, không lưu ô này vào outPath vì không thể di chuyển đến.
- Ký tự 'R': Nếu bạn gặp gương (ký tự R), gương sẽ đảo ngược hành vi di chuyển của Holmes và Watson. Nếu hành vi di chuyển của Holmes và Watson có 5 ký tự và họ đang

di chuyển ở bước đi thứ 3 trong 5 ký tự, thì sau khi đảo ngược chuỗi hành vi di chuyển, Holmes và Watson sẽ tiếp tục di chuyển ở bước đi thứ 4 của chuỗi đã bị đảo ngược.

- Ký tự 'T': Được biết rằng có chính xác 2 cửa sập trong đền thờ. Nếu Holmes và Watson di chuyển vào một cửa sập, họ sẽ được chuyển đến cửa sập còn lại. Chỉ lưu vị trí của cửa sập bước vào, không lưu vị trí của cửa sập còn lại vào outPath hay các ô đã đi qua.
- Ký tự 'H': Holmes và Watson bị dịch chuyển (theo thời gian) trở lại nơi họ đã đi qua trong quá khứ. Được biết rằng đây sẽ là nơi mà, tại thời điểm đó, Holmes và Watson đã tiêu thụ 1/3 thể lực ban đầu của họ. Nếu họ chưa tiêu thụ 1/3 thể lực ban đầu của mình, Holmes và Watson sẽ trở lại vị trí ban đầu. Sau khi quay trở lại điểm này, Holmes và Watson sẽ tiếp tục di chuyển với bước đi tiếp theo trong hành vi di chuyển hiện tại. Lưu ý rằng các ô họ đã đi qua sẽ được đặt lại sao cho khớp với thời điểm trong quá khứ. Tức là, nếu một ô nào đó đã đi qua trong quá khứ, thì khi Holmes và Watson bị dịch chuyển về quá khứ, ô đó sẽ được tính là chưa đi qua. **Về mặt hiện thực:**

- Holmes trở lại vị trí trong quá khứ
 - Bản đồ được trở lại quá khứ
 - Danh sách các ô đã đi qua được cập nhật lại cho tương ứng với quá khứ.
 - Sau khi thực hiện ứng của sự vật/sự kiện, ta mới giảm thể lực đi (nếu có). Sau đó, nếu thể lực bị mất đi một lượng 1/3, đây chính là thời điểm quá khứ mà ta cần quay trở lại nếu gặp Máy thời gian 'H'.
 - Tương tác với Bình thuốc 'P', Gương 'R': Khi quay lại quá khứ, ta vẫn tiếp tục duy trì hiệu ứng của Bình thuốc và Gương.
 - outPath, stamina, outTime sẽ giữ nguyên khi quay lại quá khứ.
- Ký tự 'P': Nếu Holmes và Watson nhặt được lọ thuốc trong đền thờ, họ sẽ được ban tặng sức mạnh huyền bí. Cụ thể, từ khoảnh khắc họ uống thuốc, họ sẽ giảm đi 1 thể lực khi đi 2 bước (thay vì 1 bước trước đó). Lưu ý rằng thể lực của Holmes và Watson sẽ chỉ giảm 1 nếu họ đi đủ 2 bước. Hiệu ứng này không cộng dồn.
 - Ký tự 'C': Holmes và Watson đã tìm thấy một loạt các biểu tượng và dấu hiệu bí ẩn được khắc trên tường trong quá trình điều tra của họ. Khi họ chạm vào những biểu tượng này, bản đồ sẽ ngay lập tức được xoay 180 độ. Không cần phải cập nhật lại outPath để tương ứng với bản đồ sau khi xoay. Tuy nhiên, phải cập nhật lại danh sách các ô đã đi qua.
 - Ký tự 'L': Trong khi điều tra đền thờ, Holmes và Watson tìm thấy một nút ẩn bên trong hang động. Khi được nhấn, các chương ngại vật sẽ tự động sáng lên bởi những ngọn đuốc gắn trên chúng trong phạm vi 1 ô vuông (bao gồm các ô chéo), phạm vi phát sáng sẽ không sáng qua phía bên kia của bản đồ. Tại thời điểm này, nếu bản đồ được thắp sáng bởi ngọn đuốc, Holmes và Watson đã tìm thấy bản đồ và cuộc điều tra kết thúc thành

công.

- Ký tự 'M': Thật không may, nếu Holmes và Watson bước lên công tắc bẫy của đèn thờ, họ sẽ triệu hồi quái vật Cthulhu và bị nó tâm trí nuốt chửng, cuộc điều tra sẽ kết thúc.

3.3 Nhiệm vụ 3: Chiếc hòm bị khóa

Sau khi điều tra nạn nhân trên bàn thờ, Holmes và Watson cũng tìm thấy một số thuốc trên bàn, khiến Holmes suy luận rằng nó đang được sử dụng như một loại thuốc ngủ để khuất phục các nạn nhân để họ có thể bị bắt cóc và buôn bán ra nước ngoài. Ngoài những loại thuốc đó, Holmes còn tìm thấy một chiếc hòm được khóa bằng khóa kết hợp 3 chữ số. Bên cạnh chiếc hòm là một lá thư của Tiến sĩ Gygas, với một ghi chú bao gồm một hình tam giác, một chuỗi số và một số được viết bằng máu. Holmes đoán rằng phải có sự liên quan giữa ba dữ kiện này để mở ổ khóa.

SV được yêu cầu viết hàm sau để giúp Sherlock mở khóa. Thông tin của hàm như sau:

- Tên hàm: **solveCrate**.
 - Tham số đầu vào:
 - **arr**: Một mảng các số nguyên dương có giá trị từ 1 đến 999.
 - **size**: Kích thước của mảng, có giá trị từ 1 đến 50.
 - **k**: Một số nguyên có giá trị nhỏ hơn **size**.
 - Yêu cầu hàm: Tạo một tam giác các mảng số với các quy tắc sau:
 - Hàng đầu tiên của tam giác sẽ là mảng gốc.
 - Bắt đầu từ hàng thứ hai trở đi, số lượng phần tử trong mảng sẽ giảm 1 so với hàng trước. Đồng thời, mỗi phần tử trong hàng này sẽ bằng tổng của hai phần tử liền kề trong hàng trước.
- Mật mã của chiếc hòm là tổng của hàng thứ k ; nếu số lớn hơn 3 chữ số, hãy lấy 3 chữ số cuối cùng.
- Kết quả trả về: Hàm trả về 3 chữ số cuối cùng của tổng hàng thứ k .

Ví dụ 4: Biết rằng $\mathbf{arr} = \{1, 2, 3, 4, 5\}$, $n = 5$. Vậy, tam giác của mảng sẽ là:

Hàng thứ 1: $\{1, 2, 3, 4, 5\}$

Hàng thứ 2: $\{3, 5, 7, 9\}$

Hàng thứ 3: $\{8, 12, 16\}$

Hàng thứ 4: $\{20, 28\}$

Hàng thứ 5: {48}

Nếu k bằng 2, hàm trả về giá trị 24.

3.4 Tạm kết

Sau khi mở chiếc hòm và tìm thấy một ghi chú có nhãn "Black Edelweiss Institute", Holmes nhớ lại rằng hoa núi Edelweiss (*Leontopodium alpinum*) là biểu tượng quốc gia của Thụy Sĩ.

Sherlock sớm nhận ra rằng có nhiều điều hơn thế nữa trong vụ án này so với những gì anh ta có thể tưởng tượng khi mới bắt đầu. Cuộc tìm kiếm câu trả lời của anh ta sẽ đưa anh ta không chỉ đến 'Lục địa' mà còn vượt Đại Tây Dương đến tận nước Mỹ, vén màn một âm mưu đen tối trên đường đi.

4 Nộp bài

Sinh viên nộp 1 tập tin: **awakened.h** trong site "Kỹ thuật lập trình (CO1027)_HK231_ALL"

Thời hạn nộp bài được công bố tại nơi nộp bài trong site nêu trên. Đến thời hạn nộp bài, đường liên kết sẽ tự động khoá nên sinh viên sẽ không thể nộp chậm. Để tránh các rủi ro có thể xảy ra vào thời điểm nộp bài, sinh viên PHẢI nộp bài trước thời hạn quy định ít nhất **một** giờ.

5 Harmony cho Bài tập lớn

Bài kiểm tra cuối kì của môn học sẽ có một số câu hỏi Harmony với nội dung của BTL. Giả sử điểm BTL mà sinh viên đạt được là **a** (theo thang điểm 10), tổng điểm các câu hỏi Harmony **b** (theo thang điểm 5). Gọi **x** là điểm của BTL sau khi Harmony, cũng là điểm BTL cuối cùng của sinh viên. Các câu hỏi cuối kì sẽ được Harmony với 50% điểm của BTL theo công thức sau:

- Nếu $a = 0$ hoặc $b = 0$ thì $x = 0$
- Nếu a và b đều khác 0 thì

$$x = \frac{a}{2} + HARM(\frac{a}{2}, b)$$

Trong đó:

$$HARM(x, y) = \frac{2xy}{x + y}$$

Sinh viên phải giải quyết BTL bằng khả năng của chính mình. Nếu sinh viên gian lận trong BTL, sinh viên sẽ không thể trả lời câu hỏi Harmony và nhận điểm 0 cho BTL.

Sinh viên **phải** chú ý làm câu hỏi Harmony trong bài kiểm tra cuối kỳ. Các trường hợp không làm sẽ tính là 0 điểm cho BTL, và bị không đạt cho môn học. **Không chấp nhận giải thích và không có ngoại lệ.**

6 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.
- Nộp nhầm bài của sinh viên khác trên tài khoản cá nhân của mình.

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.

7 Thay đổi so với phiên bản trước

- **Mục 3.2.1:** Cập nhật vị trí tiếp theo khi ra khỏi bản đồ (0,9) -> (9,0)
- **Mục 3.2.2:** Cập nhật số lượng sự vật/sự kiện trong bản đồ ngôi đền
- **Mục 3.2.2:** Làm rõ hiệu ứng của các ký tự trong bản đồ
- **Mục 3.3:** Sửa khoảng giá trị của tham số đầu vào k

————— **HẾT** —————