

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
A STUDY IN PINK - Phần 1

Phiên bản 1.0

TP. HỒ CHÍ MINH, THÁNG 02/2024

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

2 Dẫn nhập

Bài tập lớn (BTL) này được phóng tác dựa trên tập 1 mùa 1 của bộ phim Sherlock của đài BBC. Bộ phim này cũng được thực hiện dựa trên cuốn tiểu thuyết Sherlock Holmes của tác giả Sir Arthur Conan Doyle.

John Watson là một bác sĩ quân y đang nghỉ ngơi ở London sau khi bị thương tại Afghanistan. Anh suy nghĩ đến việc đổi sang căn hộ khác với giá thuê rẻ hơn thì vô tình gặp lại một người bạn cũ. Watson sau đó được bạn giới thiệu đến gặp Sherlock Holmes để cùng nhau thuê căn phòng ở số 221B đường Baker, do bà Hudson là chủ sở hữu. Từ đó, Watson bị kéo vào các vụ án đầy thử thách của Sherlock, và Sherlock có một người bạn đồng hành mới trong hành trình phá án của mình.

Vụ án gần nhất là của một người phụ nữ trong trang phục màu hồng. Vụ án này khác với các vụ án trước ở chỗ: nạn nhân dùng móng tay cào lên mặt sàn và để lại một thông điệp. Kết quả điều tra hiện trường cho thấy nạn nhân bị mất hành lý, với tài năng của mình, Sherlock đã tìm thấy hành lý bị mất. Anh đem về căn phòng số 221B đường Baker và tìm kiếm các dấu vết tiếp theo của tên tội phạm. Watson cũng trở về cùng lúc và tham gia với Sherlock.

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:

```
HP1_HP2  
EXP1_EXP2  
M1_M2  
E1_E2_E3
```

Trong đó:

- **HP₁** và **HP₂** lần lượt là chỉ số sức khỏe của Sherlock và Watson, là số nguyên dao động từ 0 đến 666. Trong bất kỳ trường hợp tính toán nào, nếu HP bị vượt quá 666 thì phải thiết lập lại bằng 666. Ngược lại, **HP** bị nhỏ hơn 0 thì phải thiết lập lại bằng 0.
- **EXP₁** và **EXP₂** lần lượt là chỉ số kinh nghiệm của Sherlock và Watson trong lúc phá án, họ tìm thấy được càng nhiều manh mối thì chỉ số kinh nghiệm sẽ càng tăng. Đây là một số nguyên nằm trong khoảng từ 0 đến 600. Khi **EXP** vượt quá 600 phải thiết lập lại bằng 600, hoặc xuống dưới 0 phải thiết lập lại bằng 0.
- **M₁** và **M₂** lần lượt là số tiền ban đầu của Sherlock và Watson, là số nguyên trong đoạn $[0, 3000]$. Khi số tiền vượt quá 3000 phải thiết lập lại bằng 3000, hoặc khi xuống dưới 0 phải thiết lập lại bằng 0.
- **E** lần lượt là mã sự kiện của các nhiệm vụ trong BTL này, là số nguyên trong đoạn $[0, 99]$.

Lưu ý:

- Trong bất kỳ trường hợp nào nếu tính toán ra số không nguyên cho **HP**, **EXP** và **M**, số đó phải được **làm tròn lên** ngay lập tức.
- Trong từng nhiệm vụ nếu có truyền vào tham số **E**, nếu **E nằm ngoài khoảng** đã cho trong mọi trường hợp của nhiệm vụ đó, hàm của nhiệm vụ sẽ không làm gì và trả về -99.

4 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 vụ án đầu tiên của Sherlock và Watson: A study in Pink, thông qua các nhiệm vụ được mô tả bên dưới.

4.1 Nhiệm vụ 1: Cuộc gặp gỡ đầu tiên (1.5 điểm)

Trong lần đầu tiên gặp gỡ, Watson được chứng kiến khả năng suy luận thiên tài của Sherlock. Sherlock đoán ra được Watson trở về từ chiến trường Afghanistan và các chi tiết khác đằng sau cuộc sống của Watson. Sinh viên được yêu cầu viết một hàm để mô tả lại quá trình mà Sherlock giải thích cho Watson những suy luận của mình. Qua quá trình này, **EXP** của Watson và của Sherlock sẽ thay đổi.

- Tên hàm: **firstMeet**.
- Tham số đầu vào:
 - **EXP₁**: chỉ số kinh nghiệm của Sherlock.
 - **EXP₂**: chỉ số kinh nghiệm của Watson.
 - **E₁**: chỉ số biểu diễn sự kiện 1.
- Kết quả trả về: Số nguyên là tổng **EXP** của Sherlock và Watson.

Lưu ý: Trong hàm ở nhiệm vụ này và các nhiệm vụ sau, các tham số biểu diễn cho các chỉ số có thể thay đổi sẽ được truyền theo kiểu tham khảo. Khi có yêu cầu cập nhật các chỉ số, sinh viên cần thực hiện cập nhật trên các biến tham khảo này. Khi đó, các biến được truyền vào cũng sẽ được cập nhật theo.

4.1.1 Trường hợp 1 (0.5 điểm)

Trong trường hợp **E₁** trong đoạn $[0, 3]$, Sherlock giải thích vì sao anh biết Watson vừa trở về từ Afghanistan. Sau đây là những thông tin mà Sherlock mô tả làm thay đổi chỉ số **EXP** của Watson:

| Thông tin | Giá trị E_1 | Quan sát của Sherlock | Tác dụng |
|-----------|---------------|---|--|
| 1 | 0 | Kiểu tóc và cách ăn nói của Watson giống như trong quân đội | Cộng 29 EXP |
| 2 | 1 | Khuôn mặt rám nắng nhưng không bị rám nắng dưới cổ tay, chứng tỏ Watson từ nước ngoài về | Cộng 45 EXP |
| 3 | 2 | Watson bước đi khập khiễng, nhưng khi gặp nhau, anh chọn đứng mà không yêu cầu ghế nên anh đã gặp vấn đề về tâm lý sau khi bị thương. Đây có thể là bị thương do hành động tại chiến trường | Cộng 75 EXP |
| 4 | 3 | Sherlock giải thích cả 3 Thông tin 1, 2, và 3 | EXP cộng thêm tương ứng với tổng 3 thông tin |

Từ những thông tin trên, Sherlock đoán ra được Watson là một bác sĩ quân y từ nước ngoài trở về, phạm vi tìm kiếm của anh giảm xuống còn 2 đất nước: Afghanistan hoặc Iraq. Lúc này quyết định của Sherlock (số nguyên D) sẽ phụ thuộc vào các yếu tố mã sự kiện E_1 và kinh nghiệm của Sherlock EXP_1 theo quan hệ như sau: $D = E_1 * 3 + EXP_1 * 7$

Nếu D là một số chẵn, Sherlock sẽ đưa ra dự đoán thiên về Afghanistan (và là một lựa chọn đúng), lúc đó EXP của anh sẽ được cộng thêm một khoảng bằng $D/200$. Ngược lại, nếu D là một số lẻ, Sherlock sẽ nghiêng về khả năng là Iraq, đây là một lựa chọn sai và EXP của anh bị giảm một khoảng bằng $D/100$.

Ví dụ 4.1

Với dữ liệu nhập là:

```
172_172
400_300
450_450
3_1_0
```

Với sự kiện $E_1 = 3$ tức là Sherlock giải thích dựa vào cả 3 thông tin

Lúc này $EXP_2 = 300 + 29 + 45 + 75 = 449$

Giá trị quyết định: $D = 3 * 3 + 400 * 7 = 2809$ là một số lẻ

Nên EXP_1 bị giảm xuống còn: $EXP_1 = 400 - (2803/100) = 371.91 \xrightarrow{\text{Làm tròn lên}} 372$

Hàm trả về $output = EXP_1 + EXP_2 = 449 + 372 = 821$

4.1.2 Trường hợp 2 (1.0 điểm)

Trong trường hợp E_1 trong đoạn $[4, 99]$, Sherlock giải thích vì sao anh biết Watson có một người anh trai. Sau đây là những thông tin mà Sherlock mô tả làm thay đổi EXP của Watson:

| Thông tin | Khoảng của E_1 | Quan sát của Sherlock | Tác dụng |
|-----------|------------------|---|---------------------------|
| 1 | [4, 19] | Watson có một chiếc điện thoại đắt tiền nhưng anh lại tìm một người ở ghép, điện thoại hẳn là do người khác tặng Watson | Cộng $(E_1/4 + 19)$ EXP. |
| 2 | [20, 49] | Điện thoại có nhiều vết xước biểu hiện rằng nó đã được đặt chung với nhiều đồ khác như chìa khoá, tiền xu. Watson sẽ không đối xử như vậy với một món đồ xa xỉ với anh. Đây là do người chủ trước gây ra với điện thoại | Cộng $(E_1/9 + 21)$ EXP. |
| 3 | [50, 65] | Trên điện thoại có khắc tên: Harry Watson, thể hiện đây là do một thành viên cũ trong gia đình đã tặng cho anh | Cộng $(E_1/16 + 17)$ EXP. |
| 4 | [66, 79] | Sherlock giải thích thông tin 1; sau khi Watson nghe xong và được cập nhật EXP_2 , nếu $EXP_2 > 200$ thì Sherlock tiếp tục giải thích thông tin 2 và Watson được cập nhật EXP_2 tương ứng. | (Như mô tả trước.) |
| 5 | [80, 99] | Sherlock giải thích thông tin 1 rồi đến thông tin 2; Watson nghe xong và được cập nhật EXP_2 , nếu $EXP_2 > 400$ thì Sherlock tiếp tục giải thích thông tin 3 và Watson được cập nhật EXP_2 tương ứng. | (Như mô tả trước.) |

Lưu ý: Nếu Watson được Sherlock giải thích cả 3 thông tin 1, 2 và 3, Watson sẽ được cộng thêm 15% của EXP đang có (sau khi đã cập nhật EXP cho cả 3 thông tin).

Sau khi Sherlock giải thích cho Watson, Watson cho biết: "Harry là tên viết tắt cho Harriet". Do vậy, Harry là *chị gái* của Watson chứ không phải *anh trai*. Bị bất ngờ trước sai lầm này, EXP của Sherlock bị giảm đi E_1 EXP.

Ví dụ 4.2

Với $EXP_1 = 500$, $EXP_2 = 450$, $E_1 = 40$. Theo thông tin 2, ta có:

$$EXP_2 = EXP_2 + (E_1/9 + 21) \approx 475.44 \xrightarrow{\text{Làm tròn lên}} 476$$

EXP_1 của Sherlock bị giảm xuống:

$$EXP_1 = EXP_1 - E_1 = 500 - 40 = 460$$

Hàm trả về giá trị:

$$\text{output} = 476 + 460 = 936$$

Ví dụ 4.3

Với $EXP_1 = 500$, $EXP_2 = 450$, $E_1 = 81$. Theo thông tin 5, EXP_2 của Watson trước tiên được cập nhật theo thông tin 1 và 2, ta có:

$$EXP_2 = EXP_2 + (E_1/4 + 19) \approx 489.25 \xrightarrow{\text{Làm tròn lên}} 490$$

$$EXP_2 = EXP_2 + (E_1/9 + 21) = 520$$

Vì $EXP_2 > 400$ nên Sherlock tiếp tục giải thích thông tin 3 và EXP_2 của Watson tiếp tục được tăng lên:

$$EXP_2 = EXP_2 + (E_1/16 + 17) = 542.06 \xrightarrow{\text{Làm tròn lên}} 543$$

Vì Watson được giải thích cả 3 thông tin 1, 2, 3, Watson được cộng thêm 15%:

$$EXP_2 = EXP_2 * 1.15 \xrightarrow{\text{Làm tròn lên}} 624.45 \xrightarrow{\text{Lớn hơn 600}} 600$$

EXP_1 của Sherlock bị giảm xuống:

$$EXP_1 = EXP_1 - E_1 = 419$$

Hàm trả về giá trị:

$$\text{output} = 600 + 419 = 1019$$

4.2 Nhiệm vụ 2: Truy tìm hành lý (2.5 điểm)

Sau lần gặp gỡ đầu tiên, Watson ngạc nhiên với khả năng suy luận thiên tài của Sherlock. Ngay ngày hôm sau, hai người cùng nhau đến xem căn hộ số 221B đường Baker của bà Hudson. Lúc này Thanh tra Thám tử Lestrade - người phụ trách các vụ tự tử gần đây đến nhờ Sherlock điều tra vụ án. Qua điều tra hiện trường, Sherlock phát hiện nạn nhân đi từ một nơi có mưa đến đây và bị mất hành lý. Sau khi tra thời tiết thì Sherlock tìm ra nơi gần nhất có mưa là Cardiff. Anh bắt đầu thử đi tìm các con đường từ Cardiff đến hiện trường xảy ra vụ án để tìm kiếm hành lý của nạn nhân.

Sinh viên được yêu cầu viết một hàm để mô tả lại quá trình tìm kiếm hành lý của Sherlock, thông tin hàm được mô tả như sau:

- Tên hàm: `traceLuggage`
- Tham số đầu vào:
 - Chỉ số sức khỏe của Sherlock HP_1 .
 - Chỉ số kinh nghiệm của Sherlock EXP_1 .
 - Số tiền của Sherlock M_1 .
 - Sự kiện E_2 .
- Kết quả trả về: $HP_1 + EXP_1 + M_1$.

Sau khi loại trừ các khả năng, Sherlock tìm thấy 3 con đường có khả năng mà tên tội phạm đã đưa nạn nhân đi và có thể hắn sẽ vất bỏ hành lý bên đường ngay khi phát hiện ra hành lý trên xe. Sherlock phải thử từng con đường để tìm ra hành lý bị mất.

Trong mỗi con đường sau đây, tùy thuộc vào máu và kinh nghiệm của Sherlock mà tạo ra những tỉ lệ tìm được vali khác nhau. Cụ thể:

4.2.1 Con đường 01

Gọi S là số chính phương gần với giá trị EXP_1 nhất. Nếu $EXP_1 \geq S$, xác suất để Sherlock tìm thấy vali trên đường này là:

$$P_1 = 100\%$$

Nếu không:

$$P_1 = (\frac{EXP_1}{S} + 80)/123$$

4.2.2 Con đường 02

Trên con đường này, Sherlock cần chi các khoản tại các sự kiện trải qua trên đường. Với số tiền M_1 của mình, Sherlock cần trải qua các sự kiện sau đây:

- Nếu $HP_1 < 200$, Sherlock sẽ dừng lại cửa hàng ven đường mua thức ăn và nước uống để hồi phục sức khỏe. Lúc này HP của Sherlock sẽ được cộng thêm 30% HP hiện có, đồng thời tiền sẽ bị trừ đi 150. Nếu HP_1 không nhỏ hơn 200, Sherlock chỉ cần mua nước uống và lúc này HP sẽ được cộng thêm 10% HP hiện có, đồng thời tiền sẽ bị trừ đi 70.
- Quảng đường cần di chuyển khá dài nên Sherlock sẽ cần thuê taxi hoặc xe ngựa. Giá tiền thuê taxi để đi hết quảng đường này là 200, còn xe ngựa 120. Nếu EXP của Sherlock < 400 , Sherlock sẽ chọn đi taxi, ngược lại sẽ đi xe ngựa. Lúc này, EXP của Sherlock sẽ tăng 13%.
- Trên quảng đường đi, Sherlock gặp một người vô gia cư và người này hứa sẽ tiết lộ manh mối đã nhìn thấy chiếc vali ở đâu cho Sherlock nghe nếu Sherlock giúp đỡ mình một ít tiền. Nếu EXP của Sherlock < 300 , Sherlock sẽ tin lời và giúp đỡ người vô gia cư này $m = 100$ và nghe người này hướng dẫn. Nếu EXP của Sherlock từ 300 trở lên, Sherlock sẽ giúp đỡ $m = 120$ và nhờ người này dẫn đường đi. Mặc dù vậy, người vô gia cư đã nhầm lẫn với một chiếc vali trống khác. EXP của Sherlock sẽ bị giảm đi 10%.

Nếu E_2 là một số lẻ, các sự kiện này sẽ tiếp tục được lặp lại cho đến khi có thời điểm tại một sự kiện hoàn thành, số tiền cần chi trả lớn hơn 50% số tiền đã có lúc bắt đầu con đường 02 này, Sherlock sẽ chọn chỉ tiếp tục đi bộ đến cuối quảng đường mà không trải qua sự kiện tiếp theo đó. Lúc này HP_1 sẽ giảm 17%, đồng thời EXP_1 tăng 17%. Ngược lại, nếu E_2 là một số chẵn, Sherlock chỉ thực hiện một lượt các hành động đó và tiếp tục đi bộ đến cuối. Trong trường hợp này, nếu M không đủ để thực hiện hết 3 hành động, Sherlock sẽ dừng thực hiện sau khi hoàn thành hành động làm cho $M_1 = 0$. EXP và HP vẫn sẽ cập nhật như với E_2 là số lẻ.

Xác suất P_2 tìm thấy vali trên tuyến đường này sẽ được tính tại lúc cuối đường và cách tính như trên con đường 01.

4.2.3 Con đường 03

Cho một mảng số cố định gồm 10 phần tử là 10 giá trị xác suất

$$P = \{32, 47, 28, 79, 100, 50, 22, 83, 64, 11\}$$

Gọi i là chỉ số giá trị xác suất P_i mà Sherlock tìm thấy vali trên con đường này (i được đánh chỉ số từ 0). Nếu E_2 là số có một chữ số, giá trị đó là giá trị của i . Nếu E_2 là số có 2 chữ số, tính tổng của 2 chữ số đó và lấy số hàng đơn vị của giá trị tổng làm giá trị cho i .

Sau khi đi hết cả 3 tuyến đường, nếu cả 3 tuyến đường mà Sherlock đã đi qua đều có xác suất là 100%, tức là Sherlock đã nhầm lẫn đâu đó và cần tính toán lại. Lúc này EXP_1 giảm đi 25%. Nếu không phải tất cả đều 100%, trung bình cộng 3 giá trị xác suất chính là xác suất cuối cùng tìm được vali. Nếu giá trị này nhỏ hơn 50%, Sherlock sẽ rất vất vả mới tìm được vali, do vậy cuối cùng, HP_1 sẽ giảm 15% và EXP_1 tăng 15%. Ngược lại nếu giá trị này lớn hơn hoặc bằng 50%, Sherlock sẽ nhanh chóng tìm được vali, do vậy cuối cùng, HP_1 sẽ giảm 10% và EXP_1 tăng 20%. (Lưu ý: EXP và HP được tính toán trên giá trị sau khi đã đi qua cả 03 tuyến đường).

Ví dụ 4.4

Với $E_2 = 39$, $HP_1 = 333$, $EXP_1 = 430$, $M_1 = 890$

Ta có: Giá trị số chính phương gần EXP_1 nhất là $S = 441$, nên:

Xác suất $P_1 = (\frac{430}{441} + 80)/123 = 0.66$

Trên con đường 02: 50% số tiền ban đầu là: $890 * 0.5 = 445$

$E_2 = 39$ là số lẻ nên:

- Do $HP_1 = 333 > 200$ nên: $M_1 = 890 - 70 = 820$
 $HP_1 = 333 * 1.1 = 366.3 \xrightarrow{\text{Làm tròn lên}} 367$
Tổng số tiền đã chi là $70 < 445$
- Do $EXP_1 = 430 > 400$ nên: $M_1 = 820 - 120 = 700$
 $EXP_1 = 430 * 1.13 = 485.9 \xrightarrow{\text{Làm tròn lên}} 486$
Tổng số tiền đã chi là $70 + 120 = 190 < 445$
- Do $EXP_1 = 486 > 300$ nên: $M_1 = 700 - 120 = 580$
 $EXP_1 = 486 * 0.9 = 437.4 \xrightarrow{\text{Làm tròn lên}} 438$
Tổng số tiền đã chi là $190 + 120 = 310 < 445$

Tiếp tục lặp lại các sự kiện:

- Do $HP_1 = 367 > 200$ nên: $M_1 = 580 - 70 = 510$
 $HP_1 = 367 * 1.1 = 403.7 \xrightarrow{\text{Làm tròn lên}} 404$
Tổng số tiền đã chi là $310 + 70 = 380 < 445$
- Do $EXP_1 = 540 > 400$ nên: $M_1 = 510 - 120 = 390$
 $EXP_1 = 438 * 1.13 = 494.94 \xrightarrow{\text{Làm tròn lên}} 495$
Tổng số tiền đã chi là $380 + 120 = 500 > 445$

Lúc này Sherlock sẽ dừng lại và chỉ đi bộ đến cuối đường

$HP_1 = 404 * 0.83 = 335.32 \xrightarrow{\text{Làm tròn lên}} 336$

$EXP_1 = 495 * 1.17 = 579.15 \xrightarrow{\text{Làm tròn lên}} 580$

Số chính phương gần EXP_1 nhất là 576 nên **Xác suất** $P_2 = 100\%$

Trên con đường thứ 03: Với $E_3 = 39$

Tổng 2 chữ số là $3 + 9 = 12$ Do đó: $i = 2$

Vậy $P_3 = P[2] = 28\%$

Xác suất trung bình: $P = \frac{P_1 + P_2 + P_3}{3} = 65\% > 50\%$

Do vậy:

$HP_1 = 336 * 0.9 = 302.4 \xrightarrow{\text{Làm tròn lên}} 303$

$EXP_1 = 580 * 1.2 = 696 \xrightarrow{\text{Lớn hơn 600}} 600$

Hàm trả về: $303 + 600 + 390 = 1293$

4.3 Nhiệm vụ 3: Đuổi theo taxi (3 điểm)

Sau khi tìm được hành lý, Sherlock đã suy luận rằng: Nạn nhân sẽ cầm theo điện thoại của mình. Chiếc điện thoại không có ở hiện trường, nhưng cũng không có trong hành lý. Vậy, nó rất có thể ở chỗ tên tội phạm. Sherlock bảo Watson hãy gửi một tin nhắn đến điện thoại của nạn nhân, Watson hãy nhắn rằng mình vừa tỉnh dậy sau khi bị ngất và không biết có chuyện gì xảy ra. Sau đó, cả hai hẹn người đang giữ điện thoại đến gặp tại một địa chỉ để lấy lại đồ.

Sau tin nhắn ấy, Sherlock chắc chắn rằng, nếu đó là tên tội phạm, hắn sẽ lo lắng khi nghe nạn nhân vẫn còn sống. Tên tội phạm sẽ đến điểm hẹn để xem tình trạng thực tế của nạn nhân. Sherlock và Watson đến một cửa hàng bên đường cách điểm hẹn khoảng 5m và cùng theo dõi. Một chiếc taxi đến dừng tại đó, người ngồi trên taxi nhìn ra ngoài với vẻ tìm kiếm. Khi người này vô tình nhìn về hướng Sherlock, chiếc xe nổ máy và rời đi. Sherlock thông thuộc rõ các con đường tại thành phố anh sống. Anh cùng Watson chạy qua các lối đi tắt và đuổi theo chiếc taxi.

Sinh viên được yêu cầu viết hàm sau để mô tả lại quá trình này. Thông tin của hàm như sau:

- Tên hàm: **chaseTaxi**
- Tham số đầu vào:
 - Các thông tin của Sherlock và Watson lần lượt là HP_1, EXP_1 và HP_2, EXP_2
 - Mã sự kiện E_3
- Yêu cầu hàm:
 - Khởi tạo một ma trận 10×10 với mỗi phần tử của mảng là một số nguyên được khởi tạo các giá trị 0. Mảng 2 chiều này biểu diễn bản đồ mà chiếc taxi đó chaya5 cùng với Sherlock và Watson đuổi theo. Giá trị của từng vị trí là điểm số kỹ năng của chiếc taxi đó tại vị trí đó.
 - Bắt đầu tại vị trí $(0,0)$, xe taxi đi theo chiều theo từng hàng. Với mỗi lần đi qua một điểm, điểm số của taxi tại điểm đó bằng $((E_3 * j) + (i * 2)) * (i - j)$ (với i là chỉ số hàng và j là chỉ số cột)
 - Ta định nghĩa đường chéo trái của một ma trận tại một vị trí $X(i, j)$ là đường đi theo hướng chéo từ một vị trí tại hàng đầu tiên hoặc cột đầu tiên sao cho đi qua X và kết thúc tại một vị trí ở hàng cuối hoặc cột cuối. Hướng chéo được định nghĩa là hướng mà chỉ số hàng và cột cùng thay đổi tuần tự 1 đơn vị (thay đổi có thể tăng hoặc giảm). Ngược lại, đường chéo phải của một ma trận tại một vị trí $X(i, j)$ là đường đi theo hướng chéo đi từ một vị trí tại hàng hoặc cột cuối cùng sao cho đi

qua X và kết thúc tại một vị trí ở hàng đầu hoặc cột đầu.

Ví dụ ta có ma trận:

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

Đường chéo trái của điểm có tọa độ $(1,0)$ bao gồm vị trí $(1,0)$ và $(2,1)$, tức là giá trị 4 và 8. Trong khi đó, đường chéo phải của điểm có tọa độ $(1,2)$ bao gồm vị trí $(1,2)$ và $(0,1)$ tức là giá trị 6 và 8 (Lưu ý: Chỉ số hàng và cột bắt đầu từ 0, hàng đi từ trên xuống, cột đi từ trái sang).

- Trong khi đó, Sherlock và Watson đi theo các lối tắt (đi theo từng cột). Ta cũng cần một ma trận để lưu điểm của họ. Điểm số của họ khi đi qua từng địa điểm này bằng **giá trị lớn nhất** ở cả đường chéo trái và đường chéo phải của chiếc taxi đã đi qua tại điểm đó (cần phải tính tất cả giá trị điểm tại từng vị trí cho chiếc taxi trước khi tính toán điểm cho Sherlock và Watson). **Nếu điểm số này là số âm, lấy giá trị tuyệt đối của nó.**
- Xe taxi sẽ gặp Sherlock và Watson tại một điểm (i, j) . Với i bằng số giá trị điểm của taxi lớn hơn $E_3 * 2$. Và j bằng số giá trị điểm của taxi nhỏ hơn $-E_3$. Nếu i hoặc j là số có 02 chữ số thì tiếp tục cộng 02 chữ số đó lại cho đến khi tổng chỉ còn 1 chữ số.
- Tại vị trí gặp nhau đó, nếu **giá trị tuyệt đối** điểm số của taxi lớn hơn 2 người họ, họ sẽ không đuổi kịp taxi. Ngược lại, họ sẽ bắt kịp được taxi.
- Nếu đuổi kịp, **EXP** và **HP** của từng người được tăng tương ứng 12% và 10%. Ngược lại sẽ bị trừ tương ứng 12% và 10%.
- Kết quả trả về: Hàm trả về số điểm lớn hơn tương ứng so giữa xe taxi với Sherlock và Watson tại điểm gặp nhau (Lưu ý: Trả về đúng giá trị âm nếu điểm số là số âm).

Ví dụ 4.5

Với $E_3 = 59$, $HP_1 = 400$, $EXP_1 = 600$, $HP_2 = 350$, $EXP_2 = 500$

Ta có ma trận điểm số tại từng vị trí đi qua của taxi là:

| | | | | | | | | | |
|------------|------------|------------|------------|------------|-------------|--------------|--------------|--------------|--------------|
| 0 | -59 | -236 | -531 | -944 | -1475 | -2124 | -2891 | -3776 | -4779 |
| 2 | 0 | -120 | -358 | -714 | -1188 | -1780 | -2490 | -3318 | -4264 |
| 8 | 63 | 0 | -181 | -480 | -897 | -1432 | -2085 | -2856 | -3745 |
| 18 | 130 | 124 | 0 | -242 | -602 | -1080 | -1676 | -2390 | -3222 |
| 32 | 201 | 252 | 185 | 0 | -303 | -724 | -1263 | -1920 | -2695 |
| 50 | 276 | 384 | 374 | 246 | 0 | -364 | -846 | -1446 | -2164 |
| 72 | 355 | 520 | 567 | 496 | 307 | 0 | -425 | -968 | -1629 |
| 98 | 438 | 660 | 764 | 750 | 618 | 368 | 0 | -486 | -1090 |
| 128 | 525 | 804 | 965 | 1008 | 933 | 740 | 429 | 0 | -547 |
| 162 | 616 | 952 | 1170 | 1270 | 1252 | 1116 | 862 | 490 | 0 |

Số giá trị dương trong ma trận lớn hơn ($E_3 * 2$) là 37

Vậy $i = 3 + 7 \rightarrow 10 \rightarrow 1 + 0 = 1$

Số giá trị âm trong ma trận nhỏ hơn ($-E_3$) là 44

Vậy $j = 4 + 4 = 8$

Vị trí gặp nhau là (1, 8)

Tại vị trí này, điểm của chiếc taxi là -3318

Tại vị trí này, điểm số của Sherlock và Watson là 660

Do $\text{abs}(-3318) > 660$ nên Sherlock và Watson sẽ không đuổi kịp taxi. Các giá trị được cập nhật lần lượt:

$$EXP_1 = EXP_1 * 0.88 = 528$$

$$HP_1 = HP_1 * 0.9 = 360$$

$$EXP_2 = EXP_2 * 0.88 = 440$$

$$HP_2 = HP_2 * 0.9 = 315$$

Hàm trả về: **-3318**

Ví dụ 4.6

Với $E_3 = 99$, $HP_1 = 400$, $EXP_1 = 600$, $HP_2 = 350$, $EXP_2 = 500$

Sau khi tính toán ma trận điểm số tại từng vị trí taxi đi qua, ta có:

Số giá trị dương trong ma trận lớn hơn ($E_3 * 2$) là 35

Vậy $i = 3 + 5 = 8$

Số giá trị âm trong ma trận nhỏ hơn ($-E_3$) là 44

Vậy $j = 4 + 4 = 8$

Vị trí gặp nhau là (8, 8)

Tại vị trí này, điểm của chiếc taxi là 0

Tại vị trí này, điểm số của Sherlock và Watson là 1442

Do $\text{abs}(0) < 1442$ nên Sherlock và Watson sẽ đuổi kịp taxi. Các giá trị được cập nhật lần lượt:

$$EXP_1 = EXP_1 * 1.12 = 672$$

$$HP_1 = HP_1 * 1.1 = 440$$

$$EXP_2 = EXP_2 * 1.12 = 560$$

$$HP_2 = HP_2 * 1.1 = 385$$

Hàm trả về: 1442

4.4 Nhiệm vụ 4: Mật khẩu hợp lệ (1.5 điểm)

Sau khi đuổi theo chiếc Taxi, Sherlock và Watson quay lại căn hộ để nghỉ ngơi. Sherlock bị chú ý bởi chiếc Laptop tìm được trong hành lý. Bên ngoài hành lý có một chiếc thẻ có địa chỉ email của nạn nhân, anh đã thử dùng email này cho tên đăng nhập, còn mật khẩu thì anh thử tất cả những mật khẩu thông thường nhưng đều không được. Trong hành lý, Sherlock còn tìm thấy một cuốn sổ ghi lại rất nhiều chuỗi ký tự khác nhau, đây có thể là mật khẩu để đăng nhập. Tuy nhiên, đa số trong số này lại vi phạm quy tắc đặt mật khẩu cho Laptop.

Vì có quá nhiều mật khẩu, SV được yêu cầu viết hàm sau để kiểm tra tính hợp lệ của một mật khẩu. Thông tin của hàm như sau:

- Tên hàm: **checkPassword**.
- Tham số đầu vào:
 - **s**: chuỗi cần kiểm tra có phải là một mật khẩu hợp lệ hay không.
 - **email**: chuỗi chứa email của nạn nhân.

- Yêu cầu hàm: Gọi *se* là chuỗi ký tự nằm trước ký tự '@' trong email của nạn nhân. Email này sẽ được đảm bảo có đúng 1 ký tự '@' và có chiều dài tối đa là 100 ký tự. Yêu cầu hợp lệ của mật khẩu là phải thỏa mãn đồng thời các điều kiện sau:
 - Có ít nhất 8 ký tự và có tối đa 20 ký tự.
 - Mỗi ký tự trong mật khẩu chỉ có thể là chữ số, hoặc chữ cái thường, hoặc chữ cái in hoa, hoặc ký tự đặc biệt. Ký tự đặc biệt là một trong các ký tự sau: '@', '#', '%', '\$', '!'.
 - Không được chứa chuỗi *se*.
 - Không được chứa nhiều hơn 2 ký tự liên tiếp.
 - Chứa ít nhất 1 ký tự đặc biệt.
- Kết quả trả về:
 - Nếu *s* là một mật khẩu hợp lệ thì trả về -10.
 - Nếu *s* có độ dài ngắn hơn độ dài tối thiểu thì trả về -1.
 - Nếu *s* có độ dài dài hơn độ dài tối đa thì trả về -2.
 - Nếu *s* có chứa *se* thì trả về $-(300 + \langle sei \rangle)$ với $\langle sei \rangle$ là vị trí xuất hiện đầu tiên của *se*.
 - Nếu *s* có chứa nhiều hơn 2 ký tự liên tiếp thì trả về $-(400 + \langle sci \rangle)$ với $\langle sci \rangle$ là vị trí đầu tiên của chuỗi đầu tiên gồm nhiều hơn 2 ký tự liên tiếp.
 - Nếu *s* không chứa ký tự đặc biệt thì trả về -5.
 - Các trường hợp còn lại trả về vị trí của ký tự đầu tiên vi phạm yêu cầu hợp lệ đã nêu trong *Yêu cầu hàm*.
- Lưu ý: nếu có nhiều điều kiện bị vi phạm thì hàm sẽ trả về điều kiện xuất hiện trước trong phần *Kết quả trả về*.

Ví dụ 4.7

Với `email = "pink@gmail.com"`, `s = "123xyz"`.

Ta có: `se = "pink"`. Chuỗi *s* có độ dài là 6, ngắn hơn độ dài tối thiểu là 8.

Hàm trả về giá trị: -1.

Ví dụ 4.8

Với `email = "pink@gmail.com"`, `s = "012345pink#pink"`.

Ta có: `se = "pink"`. Chuỗi *s* có chứa *se* ở vị trí đầu tiên xuất hiện là $\langle sei \rangle = 6$.

Hàm trả về giá trị: $-(300 + 6) = -306$.

4.5 Nhiệm vụ 5: Tìm đúng mật khẩu Laptop (1.5 điểm)

Sau khi loại bỏ các mật khẩu không hợp lệ, vẫn còn khá nhiều mật khẩu cần phải thử để tìm ra mật khẩu đúng của Laptop. Sherlock để ý rằng có các mật khẩu giống nhau xuất hiện. Sherlock đoán rằng mật khẩu xuất hiện nhiều lần nhất là mật khẩu của Laptop.

SV được yêu cầu viết hàm sau để tìm ra mật khẩu đúng của Laptop. Thông tin của hàm như sau:

- Tên hàm: **findCorrectPassword**.
- Tham số đầu vào:
 - **arr_pwds**: một mảng gồm các mật khẩu.
 - **num_pwds**: số lượng mật khẩu có trong mảng **arr_pwds**.
- Kết quả trả về: Vị trí đầu tiên của mật khẩu đúng trong mảng **arr_pwds**. Một mật khẩu đúng là mật khẩu có số lần xuất hiện trong mảng **arr_pwds** nhiều nhất và có độ dài là dài nhất trong số các mật khẩu xuất hiện nhiều nhất. Vì có thể có nhiều mật khẩu đúng nên hàm sẽ trả về vị trí đầu tiên của mật khẩu đúng.
- Ghi chú khác: Testcase sẽ đảm bảo có không quá **30** mật khẩu phân biệt trong mảng **arr_pwds**.

Ví dụ 4.9

Với

```
arr_pwds = {"123xyz", "pink#pink", "pink123", "xxxx@xxxx", "pink#pink",  
"pink123", "pink123"}.
```

Ta có:

- Mật khẩu "pink123" xuất hiện 3 lần.
- Mật khẩu "pink#pink" xuất hiện 2 lần.
- Mật khẩu "123xyz" xuất hiện 1 lần.
- Mật khẩu "xxxx@xxxx" xuất hiện 1 lần.

Do vậy, mật khẩu "pink123" có số lần xuất hiện nhiều nhất. Vị trí đầu tiên của "pink123" trong **arr_pwds** là 2.

Hàm trả về giá trị: 2.

Ví dụ 4.10

Với

```
arr_pwds = {"pink123", "xxxx@xxxx", "123xyz", "pink#pink", "pink#pink",  
"pink123"}.
```

Ta có:

- Mật khẩu "pink123" xuất hiện 2 lần và có độ dài là 7.
- Mật khẩu "pink#pink" xuất hiện 2 lần và có độ dài là 9.
- Mật khẩu "123xyz" xuất hiện 1 lần.
- Mật khẩu "xxxx@xxxx" xuất hiện 1 lần.

Hai mật khẩu "pink123" và "pink#pink" có số lần xuất hiện nhiều nhất (đều bằng 2). Nhưng mật khẩu "pink#pink" dài hơn "pink123" nên mật khẩu đúng là "pink#pink". Vị trí đầu tiên của "pink#pink" trong arr_pwds là 3. Hàm trả về giá trị: 3.

4.6 Tạm kết

Sau khi mở được laptop, trên Desktop của nạn nhân chỉ có một số phần mềm văn phòng cơ bản và 1 phần mềm định vị. Phần mềm này đã được cài đặt kết nối đến điện thoại của nạn nhân. May mắn là, tên đăng nhập cùng mật khẩu của phần mềm đều giống với thông tin này khi đăng nhập laptop. Phần mềm bắt đầu tìm kiếm, khu vực trên màn hình bắt đầu thu nhỏ phạm vi và hiển thị địa chỉ 221B phố Baker. Bà Hudson chạy lên báo một chiếc Taxi đậu đang đậu dưới nhà, tài xế nhờ bà chuyển lời: **"Taxi đặc biệt dành cho Sherlock Holmes."** Sherlock chợt hiểu ra mọi chuyện, anh bảo Watson ở lại tiếp tục tìm kiếm, anh cần ra ngoài hít thở một chút.

Sherlock sẽ làm gì để đối phó với tài xế taxi ngoài cửa, người mà khả năng cao chính là hung thủ của 4 vụ án tự tử. Liệu Watson có theo kịp Sherlock và hỗ trợ anh đánh bại tên tội phạm này. Chúng ta sẽ tiếp tục thực hiện các nhiệm vụ mới cùng Sherlock và Watson trong phần 2 của Bài tập lớn này.

Chúc các bạn làm Bài tập lớn vui vẻ!!!

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, study_in_pink1.h, study_in_pink1.cpp, và các file dữ liệu đọc mẫu. Sinh viên phải nộp 2 tập tin là study_in_pink1.h và study_in_pink1.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 study_in_pink1.cpp -I . -std=c++11
```

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

```
./main sa_tc_01_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 study_in_pink1.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 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 study_in_pink1.h, study_in_pink1.cpp để hoàn thành bài tập lớn này và đảm bảo hai yêu cầu sau:
 - Chỉ có 1 lệnh **include** trong tập tin study_in_pink1.h là **#include "main.h"** và một include trong tập tin knight.cpp là **#study_in_pink1 "knight.h"**. Ngoài ra, không cho phép có một **#include** nào khác trong các tập tin này.
 - Hiện thực các hàm được mô tả ở các Nhiệm vụ trong BTL 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: `study_in_pink1.h` và `study_in_pink1.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 (trong vòng 1 tiếng cho đến 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 Cuối kỳ với nội dung tương tự.

8 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.
- Bài 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 mã nguồn từ các công cụ có khả năng tạo ra mã nguồn mà không hiểu ý nghĩa.

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.