

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

**CUỘC TỔNG TIẾN CÔNG VÀ NỔI DẬY
MÙA XUÂN NĂM 1975
Phần 1: Chiến dịch Tây Nguyên**

Phiên bản 1.2

ĐẶC TẢ BÀI TẬP LỚN

Phiên bản 1.2

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 lấy cảm hứng nhân kỷ niệm 50 năm thành công của cuộc tổng tiến công và nổi dậy mùa Xuân 1975 của quân và dân ta, mang đến chiến thắng giải phóng hoàn toàn miền Nam, thống nhất đất nước.

Sau hiệp định Paris năm 1973, Mỹ rút quân khỏi miền Nam nhưng vẫn tiếp tục viện trợ cho chính quyền Sài Gòn. Âm mưu của đế quốc Mỹ là sử dụng Quân lực Việt Nam Cộng hòa để thực hiện chủ nghĩa thực dân mới ở miền Nam Việt Nam, biến miền Nam thành thuộc địa kiểu mới. Tháng 7/1973, Hội nghị Trung ương Đảng lần thứ 21 đã khẳng định con đường cách mạng miền Nam vẫn là con đường bạo lực và đề ra các nhiệm vụ lớn cho hai miền Nam - Bắc. Tháng 7/1974, Đảng ta chỉ đạo Bộ Tổng tham mưu Quân đội Nhân dân Việt Nam xây dựng kế hoạch chiến lược giải phóng miền Nam trong hai năm 1975 - 1976, nếu thời cơ đến sẽ giải phóng miền Nam trong năm 1975 [2]. Cuối năm 1974 - đầu năm 1975, Đảng ta nhận định thời cơ giải phóng miền Nam đã đến.

Cuộc tổng tiến công và nổi dậy mùa Xuân 1975 bắt đầu vào ngày 04/03, với 3 chiến dịch quan trọng. Trong đó, mở màn là chiến dịch Tây Nguyên lịch sử, chiến thắng quan trọng tạo thời cơ và tiền đề cho Chiến dịch Giải phóng Huế - Đà Nẵng và chiến dịch Hồ Chí Minh [3].

Mục tiêu của chiến dịch Tây Nguyên là tiêu diệt một bộ phận quan trọng sinh lực địch, giải phóng các tỉnh Nam Tây Nguyên (Đắk Lắk, Phú Bổn, Quảng Đức), thực hiện chia cắt chiến lược và tạo thế chiến lược mới trên chiến trường toàn miền Nam [1].

Chỉ vỏn vẹn trong khoảng 1 tháng (03/1975), chiến dịch Tây Nguyên đã thắng lợi. Trong BTL đầu tiên này, lấy cảm hứng từ diễn biến cuộc chiến của chiến dịch Tây Nguyên, sinh viên sẽ cùng sử dụng lập trình để tái hiện một số diễn biến chính của chiến dịch.

3 Dữ liệu nhập

3.1 Mô tả 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 chứa các thông tin như sau:

```
[LF1_0, LF1_1, ..., LF1_16]
[LF2_0, LF2_1, ..., LF2_16]
EXP1 EXP2
T1 T2
E
```

Trong đó:

- **LF₁** và **LF₂**: Mỗi giá trị trong hai mảng này lần lượt biểu diễn số lượng quân nhân theo từng cấp bậc tại hai quân đoàn quan trọng trong chiến dịch Tây Nguyên. Cả hai mảng đều có 17 phần tử, mỗi phần tử ứng với một cấp bậc từ thấp nhất đến cao nhất: **Binh nhì, Binh nhất, Hạ sĩ, Trung sĩ, Thượng sĩ, Thiếu úy, Trung úy, Thượng úy, Đại úy, Thiếu tá, Trung tá, Thượng tá, Đại tá, Thiếu tướng, Trung tướng, Thượng tướng, Đại tướng**.

- [LF₁₀, LF₁₁, ..., LF₁₁₆]: Số lượng quân nhân theo từng cấp tại quân đoàn 1.
- [LF₂₀, LF₂₁, ..., LF₂₁₆]: Số lượng quân nhân theo từng cấp tại quân đoàn 2.

Mỗi phần tử trong mảng là số nguyên nằm trong khoảng [0, 1000]. Nếu có phần tử vượt quá 1000, thì phải thiết lập lại giá trị của phần tử đó bằng 1000. Nếu giảm xuống dưới 0, phải thiết lập lại bằng 0.

- **EXP₁** và **EXP₂**: Lần lượt là mức độ kinh nghiệm chiến đấu của các đơn vị quân ta tại hai quân đoàn. Đây là một số nguyên nằm trong khoảng [0, 600]. Nếu vượt quá 600, phải thiết lập lại bằng 600. Nếu nhỏ hơn 0, phải thiết lập lại bằng 0.
- **T₁** và **T₂**: Lần lượt là nguồn tiếp tế lương thực, vũ khí và đạn dược cho hai quân đoàn. Đây là số nguyên trong khoảng [0, 3000]. Nếu vượt quá 3000, phải thiết lập lại bằng 3000.

Nếu nhỏ hơn 0, phải thiết lập lại bằng 0.

- **E**: E lần lượt là mã sự kiện của các nhiệm vụ trong BTL này, là số nguyên trong đoạn [0, 99]. Nếu vượt quá 99, phải thiết lập lại bằng 99. Nếu nhỏ hơn 0, phải thiết lập lại bằng 0.

Ví dụ minh họa:

```
[100,80,60,40,20,10,5,2,1,0,0,0,0,0,0,0,0]
[120,90,70,50,30,15,10,5,3,2,1,0,0,0,0,0,0]
400_350
2500_2800
99
```

3.2 Nhiệm vụ 0: Đọc dữ liệu đầu vào (1.0 điểm)

Trước khi thực hiện các nhiệm vụ khác trong chiến dịch Tây Nguyên, quân đội cần thu thập và xử lý thông tin về lực lượng hiện có, kinh nghiệm chiến đấu, nguồn tiếp tế và các sự kiện tác động. Nhiệm vụ đầu tiên của sinh viên là xây dựng một chức năng đọc và xử lý dữ liệu từ tệp tin đầu vào.

3.2.1 Mô tả hàm

- **Tên hàm:** readFile
- **Tham số đầu vào:**
 - **filename**: Tên tệp tin chứa dữ liệu đầu vào.
 - **LF1[17]**: Mảng 1D chứa số lượng quân nhân theo từng cấp bậc tại quân đoàn 1.
 - **LF2[17]**: Mảng 1D chứa số lượng quân nhân theo từng cấp bậc tại quân đoàn 2.
 - **EXP1, EXP2**: Kinh nghiệm chiến đấu của hai quân đoàn.
 - **T1, T2**: Nguồn tiếp tế của hai quân đoàn.
 - **E**: Mã sự kiện tác động.
- **Kết quả trả về:**
 - **true**: Nếu tệp tin được mở và đọc thành công.
 - **false**: Nếu tệp tin không tồn tại hoặc có lỗi khi đọc dữ liệu.

3.2.2 Mô tả chi tiết

Sinh viên cần hiện thực hàm `readFile` để đọc nội dung từ tệp tin đầu vào, lưu trữ thông tin vào mảng ký tự trước khi phân tích và gán giá trị vào các biến thích hợp. Các yêu cầu cụ thể:

- Đọc toàn bộ nội dung của tệp tin đầu vào và lưu từng dòng vào mảng ký tự 2 chiều `char data[MAX_LINES][MAX_LINE_LENGTH]`.
- Sinh viên cần phân tách dữ liệu từ mảng ký tự và gán giá trị tương ứng vào các biến `LF1`, `LF2`, `EXP1`, `EXP2`, `T1`, `T2`, và `E`.
- Nếu số lượng dòng dữ liệu không đủ hoặc có lỗi định dạng, chương trình phải trả về `false`.
- Cần điều chỉnh các giá trị của `LF1`, `LF2`, `EXP1`, `EXP2`, `T1`, `T2`, và `E` theo điều kiện đã nêu ở mục 3.1.

3.2.3 Ví dụ minh họa

Giả sử tệp tin đầu vào có nội dung:

```
[100,80,60,40,20,10,5,2,1,0,0,0,0,0,0,0,0]
[120,90,70,50,30,15,10,5,3,2,1,0,0,0,0,0,0]
400_350
2500_2800
99
```

Sau khi đọc tệp tin và xử lý, hàm `readFile` phải gán giá trị tương ứng:

- **LF1:** [100,80,60,40,20,10,5,2,1,0,0,0,0,0,0,0,0]
- **LF2:** [120,90,70,50,30,15,10,5,3,2,1,0,0,0,0,0,0]
- **EXP1:** 400, **EXP2:** 350
- **T1:** 2500, **T2:** 2800
- **E:** 99

4 Nhiệm vụ

Trong chiến dịch Tây Nguyên, quân đội ta thực hiện 5 nhiệm vụ quan trọng để đạt được thắng lợi hoàn toàn. Sinh viên được yêu cầu xây dựng một chương trình giả lập để mô phỏng lại một

số nhiệm vụ chính trong quá trình chiến đấu và giải phóng Tây Nguyên của quân đội ta. Các nhiệm vụ cụ thể được mô tả như sau:

4.1 Nhiệm vụ 1: Tập kết lực lượng (1.0 điểm)

Trước khi tiến hành chiến dịch Tây Nguyên, quân đội ta cần tập hợp đủ quân số tại hai quân đoàn quan trọng. Trong nhiệm vụ này, lực lượng quân đội tại mỗi quân đoàn được xác định dựa trên số lượng binh sĩ của các cấp bậc khác nhau. Mỗi cấp bậc đóng góp một giá trị sức mạnh tương ứng vào tổng sức mạnh lực lượng quân đội **LF** theo công thức tính trọng số.

4.1.1 Mô tả hàm

- **Tên hàm:** `gatherForces`
- **Tham số đầu vào:**
 - **LF1[17]**: Mảng 1D gồm 17 phần tử đại diện cho số lượng quân nhân của từng cấp bậc tại quân đoàn 1.
 - **LF2[17]**: Mảng 1D gồm 17 phần tử đại diện cho số lượng quân nhân của từng cấp bậc tại quân đoàn 2.
- **Kết quả trả về:**
 - Hàm trả về **Tổng sức mạnh lực lượng quân đội**.

4.1.2 Mô tả chi tiết

Giá trị **Sức mạnh lực lượng quân đội** được xác định bởi số lượng quân nhân của từng cấp bậc, với mỗi cấp bậc có một trọng số sức mạnh riêng, thể hiện mức độ đóng góp của cấp bậc đó vào tổng sức mạnh lực lượng quân đội. Trọng số cụ thể của từng cấp bậc được xác định như sau:

Index	Cấp bậc	Trọng số	Vai trò và mô tả
0	Binh nhì	1	Chiến sĩ mới nhập ngũ, ít kinh nghiệm chiến đấu.
1	Binh nhất	2	Có thời gian phục vụ lâu hơn, kinh nghiệm thực chiến cao hơn binh nhì.
2	Hạ sĩ	3	Chỉ huy nhóm nhỏ, có thể hướng dẫn chiến sĩ cấp dưới.
3	Trung sĩ	4	Có kinh nghiệm, đóng vai trò hỗ trợ chỉ huy trung đội.
4	Thượng sĩ	5	Hỗ trợ chỉ huy đại đội, có thể tạm quyền trong một số trường hợp.
5	Thiếu úy	7	Sĩ quan cấp thấp nhất, chỉ huy trung đội, đóng vai trò quan trọng trong các trận đánh nhỏ.
6	Trung úy	8	Chỉ huy trung đội hoặc đại đội nhỏ, có kinh nghiệm tổ chức chiến thuật.
7	Thượng úy	9	Giữ vai trò lãnh đạo quan trọng trong các đơn vị cấp đại đội.
8	Đại úy	10	Chỉ huy đại đội, tham gia lập kế hoạch tác chiến ở cấp tiểu đoàn.
9	Thiếu tá	12	Chỉ huy tiểu đoàn, có trách nhiệm chiến thuật lớn hơn.
10	Trung tá	15	Chỉ huy trung đoàn hoặc tham mưu cấp cao.
11	Thượng tá	18	Giữ vai trò quan trọng trong chỉ huy trung đoàn hoặc cấp sư đoàn.
12	Đại tá	20	Chỉ huy sư đoàn, tham gia chiến lược cấp quân khu.
13	Thiếu tướng	30	Lãnh đạo quân đoàn, có quyền điều động lực lượng lớn trong chiến dịch.
14	Trung tướng	40	Chỉ huy cấp quân khu, có trách nhiệm quyết định các chiến dịch lớn.
15	Thượng tướng	50	Cấp chỉ huy tối cao trong chiến dịch, tham gia chỉ đạo cấp quốc gia.
16	Đại tướng	70	Tổng tư lệnh tối cao, có quyền quyết định toàn bộ chiến dịch.

Bảng 1: Sức mạnh lực lượng quân đội theo cấp bậc

Sức mạnh lực lượng quân đội của mỗi quân đoàn được tính bằng công thức:

$$\mathbf{LF}_1 = \sum_{i=0}^{16} (\mathbf{LF1}[i] \times \text{trọng số}[i])$$

$$\mathbf{LF}_2 = \sum_{i=0}^{16} (\mathbf{LF2}[i] \times \text{trọng số}[i])$$

Từ đó, **Tổng sức mạnh lực lượng quân đội** của 2 quân đoàn được xác định bằng công thức:

$$\mathbf{LF} = \mathbf{LF}_1 + \mathbf{LF}_2$$

4.1.3 Ví dụ minh họa

Ví dụ 4.1

Với dữ liệu nhập:

```
[200, 150, 100, 80, 50, 30, 20, 10, 5, 2, 1, 1, 1, 1, 0, 0, 0]
[250, 200, 150, 100, 80, 50, 30, 20, 10, 5, 2, 1, 1, 1, 1, 0, 0]
300_300
3000_3000
99
```

Sức mạnh lực lượng quân đội của quân đoàn 1:

$$\mathbf{LF}_1 = (200 \times 1) + (150 \times 2) + (100 \times 3) + \dots + (0 \times 70) = 1987$$

Sức mạnh lực lượng quân đội của quân đoàn 2:

$$\mathbf{LF}_2 = (250 \times 1) + (200 \times 2) + (150 \times 3) + \dots + (0 \times 70) = 2968$$

Tổng sức mạnh lực lượng quân đội: $\mathbf{LF} = 1987 + 2968 = 4955$

4.2 Nhiệm vụ 2: Đánh nghi binh (2.0 điểm)

Trong chiến dịch này, quân đội ta đã sử dụng chiến thuật **nghi binh** để đánh lừa kẻ địch, giúp đánh chiếm các mục tiêu chính mà không gặp phải sự kháng cự mạnh mẽ từ đối phương.

Để đánh lạc hướng quân đội kẻ địch, quân ta đã tạo ra các tín hiệu giả, khiến địch tưởng rằng

cuộc tấn công chính sẽ diễn ra ở **phía Bắc Tây Nguyên**, trong khi thực tế, mục tiêu chính lại nằm ở **phía Nam Tây Nguyên - Buôn Ma Thuột**.

Các khu vực nghi binh quan trọng [1]:

- **Kon Tum:**

- Quân ta tạo ra hoạt động di chuyển quân giả để địch nghĩ rằng ta sẽ tấn công Kon Tum.
- Thực hiện phát tín hiệu liên lạc vô tuyến, giả vờ chuẩn bị tấn công.

- **Pleiku:**

- Bộ đội xây dựng các công sự, vận chuyển lương thực và quân nhu giả để địch tin rằng sẽ có một cuộc tấn công lớn vào Pleiku.
- Sử dụng truyền thông và dân quân địa phương để lan truyền thông tin về một cuộc tấn công vào khu vực này.

- **Gia Lai (Thanh An, đường 19):**

- Sư đoàn 968 được lệnh hành quân từ Nam Lào về Tây Nguyên, thay thế lực lượng chủ lực 320 và 10 để đánh nghi binh tại Kon Tum - Pleiku.
- Lệnh đánh pháo binh vào Pleiku để giữ chân quân địch.

Mục tiêu quan trọng nhất của chiến dịch là **Buôn Ma Thuột**, bởi đây là trung tâm phòng thủ chiến lược của quân đội chính quyền Sài Gòn ở Tây Nguyên. Quân đội ta đã tập trung lực lượng lớn để đánh chiếm khu vực này.

Các mục tiêu chính cần đánh chiếm [1]:

- **Buôn Ma Thuột (mục tiêu quan trọng nhất):**

- Trung tâm hành chính, quân sự và hậu cần lớn nhất của chính quyền Sài Gòn ở Tây Nguyên.
- Nếu chiếm được Buôn Ma Thuột, toàn bộ hệ thống phòng thủ của địch ở Tây Nguyên sẽ sụp đổ.

- **Đức Lập (hướng Tây Buôn Ma Thuột):**

- Là căn cứ phòng thủ của ngõ phía Tây của Buôn Ma Thuột.
- Được Sư đoàn 10 tấn công trước để cắt đường tiếp viện từ hướng này.

- **Đắk Lắk (xung quanh Buôn Ma Thuột):**

- Cắt đường tiếp tế của quân đội Sài Gòn từ phía Nam và Đông Buôn Ma Thuột.

– Cô lập quân đội Sài Gòn tại thị xã Buôn Ma Thuột.

- **Quốc lộ 21 và Quốc lộ 14 (Khóa chặt Tây Nguyên):**

- Ngày 4/3/1975, Trung đoàn 25 chặn Quốc lộ 21 từ Đắk Lắk xuống Khánh Hòa, ngăn chặn quân địch rút lui.
- Ngày 6-8/3/1975, Sư đoàn 320 đánh chặn Quốc lộ 14 từ Pleiku đi Buôn Ma Thuột, cô lập chiến trường.

- **Sân bay Hòa Bình, Kho Mai Hắc Đế:**

- Tấn công ngay từ 2h sáng ngày 10/3/1975 để cắt đường không viện trợ của địch.

Từ những thông tin trên, ta có 2 bảng dữ liệu sau:

ID	Mục tiêu nghi binh	Tên tiếng Anh
0	Kon Tum	Kon Tum
1	Pleiku	Pleiku
2	Gia Lai	Gia Lai

Bảng 2: Danh sách các mục tiêu nghi binh

ID	Mục tiêu chính cần đánh chiếm	Tên tiếng Anh
3	Buôn Ma Thuột	Buon Ma Thuot
4	Đức Lập	Duc Lap
5	Đắk Lắk	Dak Lak
6	Quốc lộ 21	National Route 21
7	Quốc lộ 14	National Route 14

Bảng 3: Danh sách các mục tiêu chính cần đánh chiếm

4.2.1 Nhiệm vụ 2.1: Nghi binh

Trong nhiệm vụ này, chúng ta sẽ thực hiện chiến thuật **nghi binh** nhằm đánh lừa đối phương về mục tiêu thực sự mà quân đội ta sẽ tấn công. Cụ thể, phương pháp nghi binh được thực hiện bằng cách **mã hóa** thông tin về mục tiêu đánh chiếm, khiến kẻ thù hiểu sai hướng tấn công của ta. Nghệ thuật nghi binh trong quân sự là một nội dung rất lớn, trong BTL này ta giả sử các phương pháp mã hoá đơn giản được sử dụng cho phù hợp với khuôn khổ của BTL.

Quy tắc mã hóa nghi binh:

- Mỗi mục tiêu nghi binh được thể hiện dưới dạng một chuỗi gồm:

- **Tên tiếng Anh của mục tiêu nghi binh:** Có thể được viết nguyên bản hoặc được cách điệu bằng cách thêm chữ số hoặc thay đổi một số ký tự, nhưng vẫn đảm bảo khi kẻ địch nhìn vào có thể nhận diện được mục tiêu gốc.
- Từ 1 đến 3 số nguyên mã hóa cho **ID của mục tiêu chính cần đánh chiếm**.
- Các số nguyên mã hóa cho ID có thể xuất hiện tại **bất kỳ vị trí nào** trong chuỗi. Nếu trong chuỗi có từ 2 trở lên, thì các số nguyên này sẽ không ở cạnh nhau.
- Mỗi số nguyên trong chuỗi là một số nguyên dương thuộc khoảng $[0, 100]$.

Quy tắc giải mã ID:

- Nếu có đúng **1 số nguyên** và số nguyên đó $\in [3, 7]$, số nguyên đó chính là **ID của mục tiêu chính**.
- Nếu có đúng **2 số nguyên**, ID của mục tiêu chính được tính bằng:

$$ID = [(x_1 + x_2) \bmod 5] + 3$$

- Nếu có đúng **3 số nguyên**, ID của mục tiêu chính được xác định theo số **lớn nhất** trong ba số, sử dụng công thức:

$$ID = [(\max(x_1, x_2, x_3)) \bmod 5] + 3$$

- Nếu không có số nguyên hoặc có nhiều hơn 3 số nguyên, thì hàm sẽ trả về "INVALID".
- Trong trường hợp có 1 số nguyên, nếu số nguyên đó là ID của mục tiêu nghi binh, thì hàm trả về "DECOY". Trong trường hợp số nguyên đó không phải là mục tiêu nghi binh và cũng không phải là mục tiêu chính cần đánh chiếm, thì hàm trả về "INVALID".

Mô tả hàm giải mã

- **Tên hàm:** determineRightTarget
- **Tham số đầu vào:**
 - **target:** Chuỗi ký tự nghi binh chứa tên mục tiêu giả kèm theo ID của mục tiêu chính.
- **Giả sử:** ID là một số nguyên thuộc khoảng $[0, 100]$.
- **Kết quả trả về:**
 - Tên tiếng Anh của mục tiêu chính cần đánh chiếm.
 - "DECOY" hoặc "INVALID" trong các trường hợp được nêu ở phần **Quy tắc giải mã ID**.

Ví dụ minh họa

Ví dụ 4.2

Trường hợp 1: Có đúng 1 số

Chuỗi đầu vào có đúng **1 số nguyên** và số nguyên đó $\in [3, 7]$. Vì vậy, ID của mục tiêu chính chính là số đó.

Kon3_Tum

Kết quả giải mã:

- Kon3 Tum \Rightarrow Quân địch nghĩ rằng mục tiêu tấn công là **Kon Tum**, nhưng thực chất, mục tiêu chính cần đánh chiếm là **Buôn Ma Thuột** (ID = 3).
- Hàm trả về "**Buon Ma Thuot**".

Ví dụ 4.3

Trường hợp 2: Có đúng 2 số

Chuỗi đầu vào có hai số, ta tính ID theo công thức:

$$ID = [(x_1 + x_2) \bmod 5] + 3$$

Ple9i_ku5

Tính toán ID:

$$ID = [(9 + 5) \bmod 5] + 3 = [14 \bmod 5] + 3 = 4 + 3 = 7$$

Kết quả giải mã:

- Ple9iku5 \Rightarrow Quân địch nghĩ rằng mục tiêu tấn công là **Pleiku**, nhưng thực chất, mục tiêu chính cần đánh chiếm là **Quốc lộ 14** (ID = 7).
- Hàm trả về "**National Route 14**".

Ví dụ 4.4

Trường hợp 3: Có đúng 3 số

Chuỗi đầu vào có ba số, ID được tính theo công thức:

$$ID = [(\max(x_1, x_2, x_3)) \bmod 5] + 3$$

Ple1Ku4Nat9

Tính toán ID:

$$ID = [(\max(1, 4, 9)) \bmod 5] + 3 = [9 \bmod 5] + 3 = 4 + 3 = 7$$

Kết quả giải mã:

- Ple1Ku4Nat9 \Rightarrow Quân địch nghĩ rằng mục tiêu tấn công là **Pleiku**, nhưng thực chất, mục tiêu chính cần đánh chiếm là **Quốc lộ 14** (ID = 7).
- Hàm trả về "**National Route 14**".

Ví dụ 4.5

Trường hợp 4: Không có số nguyên hoặc có nhiều hơn 3 số

Nếu không có số nguyên hoặc có nhiều hơn 3 số nguyên trong chuỗi, kết quả sẽ là "INVALID".

KonTum

Kết quả giải mã:

- Không có số nguyên dương trong chuỗi.
- Hàm trả về "**INVALID**".

Buon1Ma3Thuot5-8

Kết quả giải mã:

- Có 4 số nguyên dương trong chuỗi.
- Hàm trả về "**INVALID**".

4.2.2 Nhiệm vụ 2.2: Giải mã

Để đảm bảo tính bí mật và bất ngờ, thông tin về mục tiêu tấn công tiếp theo của quân ta được mã hóa trong các thông điệp mật. Do đó, một trong những nhiệm vụ quan trọng đặt ra là giải mã các thông điệp này để xác định chính xác mục tiêu tấn công tiếp theo.

- **Tên hàm:** `decodeTarget`
- **Tham số đầu vào:**
 - **message:** Chuỗi ký tự mã hóa chứa thông tin về mục tiêu chính cần đánh chiếm.
 - **EXP1:** Kinh nghiệm tình báo của quân ta tại quân đoàn 1.
 - **EXP2:** Kinh nghiệm tình báo của quân ta tại quân đoàn 2.
- **Yêu cầu của hàm:**
 - Giải mã **message** bằng một trong hai phương pháp: **Caesar Cipher** hoặc **Đảo ngược chuỗi**.
 - Sau khi giải mã, kiểm tra xem chuỗi thu được có khớp với **Danh sách mục tiêu chính cần đánh chiếm** không.
 - Nếu có, trả về tên mục tiêu đã giải mã thành công. Nếu không, trả về "INVALID".

Quy tắc lựa chọn phương pháp giải mã:

- Nếu $\text{EXP}_1 \geq 300$ và $\text{EXP}_2 \geq 300$, thì sử dụng phương pháp **Caesar Cipher**.
- Nếu $\text{EXP}_1 < 300$ hoặc $\text{EXP}_2 < 300$, thì sử dụng phương pháp **Đảo ngược chuỗi**.

Phương pháp giải mã:

1. Caesar Cipher:

- Độ dịch chuyển (**shift**) được tính theo công thức:

$$\text{shift} = (\text{EXP}_1 + \text{EXP}_2) \bmod 26$$

- Nếu chuỗi **message** chứa bất kỳ ký tự nào không phải chữ cái (A-Z, a-z) và cũng không phải số nguyên trong khoảng 0-9, thì chương trình phải trả về "INVALID" ngay lập tức.
- Giải mã từng ký tự trong **message** theo quy tắc:
 - Nếu ký tự là chữ cái viết thường (a-z):

$$\text{Ký tự mới} = (\text{Ký tự cũ} - a' + \text{shift}) \bmod 26 + a'$$

– Nếu ký tự là chữ cái viết hoa (A-Z):

$$\text{Ký tự mới} = (\text{Ký tự cũ} - A' + \text{shift}) \bmod 26 + A'$$

– Nếu ký tự là khoảng trắng (' ') hoặc số nguyên từ 0 đến 9, giữ nguyên.

- Kết quả sau khi giải mã sẽ được kiểm tra với **Danh sách mục tiêu chính cần đánh chiếm**:

– Nếu khớp, trả về tên mục tiêu.

– Nếu không, trả về "INVALID".

2. **Đảo ngược chuỗi**: Đảo ngược toàn bộ chuỗi ký tự đầu vào.

Ví dụ minh họa:

Ví dụ 4.6

Trường hợp 1: Sử dụng Caesar Cipher

Dữ liệu đầu vào:

- **EXP1** = 350, **EXP2** = 400
- Chuỗi mã hóa: "Heo Peo"

Bước giải mã:

$$\text{Dịch tiến} = (350 + 400) \bmod 26 = 750 \bmod 26 = 22$$

Dịch tiến từng ký tự (trừ khoảng trắng):

"H" → "D"

"e" → "a"

"o" → "k"

" " → " "

"P" → "L"

"e" → "a"

"o" → "k"

Kết quả giải mã: **"Dak Lak"**

Đối chiếu với danh sách mục tiêu: Tìm thấy trong danh sách mục tiêu chính ⇒ Hàm trả về "Dak Lak".

Ví dụ 4.7

Trường hợp 2: Sử dụng Đảo ngược chuỗi

Dữ liệu đầu vào:

- $EXP1 = 150$, $EXP2 = 320$
- Chuỗi mã hóa: "Pal cUd"

Bước giải mã:

"Pal cUd" \rightarrow "dUc laP"

Đối chiếu với danh sách mục tiêu: Tìm thấy trong danh sách mục tiêu chính \Rightarrow Hàm trả về "Duc Lap".

4.3 Nhiệm vụ 3: Kiểm soát hậu cần (2.0 điểm)

Trong chiến dịch Tây Nguyên, việc duy trì **nguồn cung ứng hậu cần** bao gồm lương thực, vũ khí và đạn dược có vai trò quan trọng trong việc đảm bảo sức chiến đấu của quân đội ta. Để quản lý hiệu quả nguồn tiếp tế tại các quân đoàn, quân đội cần thực hiện phân bổ tài nguyên hợp lý dựa trên nhu cầu thực tế và mức độ tiêu hao trong chiến đấu.

Quân đội cần triển khai một hệ thống quản lý hậu cần để theo dõi lượng tiếp tế hiện có tại **hai quân đoàn** và điều chỉnh hợp lý dựa trên ba yếu tố quan trọng:

- **Lực lượng quân đội (LF_1, LF_2)**: Quân đoàn có quân số lớn hơn sẽ cần nhiều tiếp tế hơn.
- **Mức độ kinh nghiệm chiến đấu (EXP_1, EXP_2)**: Đơn vị có kinh nghiệm cao có thể sử dụng nguồn lực hiệu quả hơn.
- **Sự kiện lịch sử (E)**: Một số sự kiện có thể làm thay đổi lượng tiếp tế của từng quân đoàn.

4.3.1 Mô tả hàm

- Tên hàm: `manageLogistics`
- Tham số đầu vào:
 - LF_1, LF_2 : Số lượng quân đội tại hai quân đoàn.
 - EXP_1, EXP_2 : Mức độ kinh nghiệm của hai quân đoàn.

- T_1, T_2 : Lượng tiếp tế hiện có tại hai quân đoàn (**truyền theo tham chiếu**, cho phép cập nhật giá trị của biến).
- E : Mã sự kiện lịch sử ảnh hưởng đến hậu cần.

- **Kết quả trả về:**

- Hàm không trả về giá trị, nhưng cập nhật trực tiếp hai biến T_1 và T_2 sau khi điều chỉnh.
- Tất cả các giá trị T_1, T_2 đều phải đảm bảo nằm trong khoảng $[0, 3000]$. Nếu phép tính trả về một số không nguyên, giá trị đó phải được làm tròn lên ngay lập tức.

4.3.2 Mô tả chi tiết

Tiếp tế sẽ được điều chỉnh theo công thức:

$$\Delta T_1 = \left(\frac{LF_1}{LF_1 + LF_2} \times (T_1 + T_2) \right) \times \left(1 + \frac{EXP_1 - EXP_2}{100} \right)$$

$$\Delta T_2 = (T_1 + T_2) - \Delta T_1$$

$$T_1 = T_1 + \Delta T_1$$

$$T_2 = T_2 + \Delta T_2$$

Quy tắc điều chỉnh theo sự kiện lịch sử: Mã sự kiện lịch sử ảnh hưởng đến hậu cần được chia thành các khoảng giá trị sau:

- Nếu $E = 0$: Không có sự kiện đặc biệt, áp dụng công thức trên.
- Nếu $E \in [1, 9]$:
 - Quân đoàn 1 mất $(E \times 1\%) \times T_1$ tiếp tế.
 - Quân đoàn 2 mất $(E \times 0.5\%) \times T_2$ tiếp tế.
- Nếu $E \in [10, 29]$:
 - Mỗi quân đoàn nhận thêm $(E \times 50)$ đơn vị tiếp tế.
- Nếu $E \in [30, 59]$:
 - Quân đoàn 1 được tăng cường $(E \times 0.5\%) \times T_1$ tiếp tế.

- Quân đoàn 2 chỉ được tăng cường $(E \times 0.2\%) \times T_2$ tiếp tế.
- Nếu $E \in [60, 99]$: Đường tiếp tế bị gián đoạn, tiếp tế không được điều chỉnh.

4.3.3 Ví dụ minh họa

Ví dụ 4.8

Trường hợp 1: Sự kiện làm giảm tiếp tế

```
[250,200,150,100,80,50,30,20,10,5,2,1,1,1,1,0,0]
[300,250,200,150,100,80,50,30,20,10,5,2,1,1,1,0,0]
400_450
2800_3000
5
```

Do $E = 5$ thuộc khoảng $[1, 9]$, ta giảm tiếp tế:

$$T_1 = 2800 - (5 \times 1\%) \times 2800 = 2660$$

$$T_2 = 3000 - (5 \times 0.5\%) \times 3000 = 2925$$

Ví dụ 4.9

Trường hợp 2: Sự kiện tăng cường hậu cần

```
[200,180,150,120,90,60,40,30,20,10,5,2,1,1,0,0,0]
[220,190,160,130,100,70,50,30,20,10,5,2,1,1,0,0,0]
350_380
2600_2800
20
```

Do $E = 20$ thuộc khoảng $[10, 29]$, mỗi quân đoàn nhận thêm $20 \times 50 = 1000$ đơn vị:

$$T_1 = \min(2600 + 1000, 3000) = 3000$$

$$T_2 = \min(2800 + 1000, 3000) = 3000$$

4.4 Nhiệm vụ 4: Lập kế hoạch tấn công (2.0 điểm)

Trong chiến dịch Tây Nguyên, quân đội ta cần lập kế hoạch tấn công chiến lược để tiêu diệt toàn bộ cứ điểm của đối phương tại các khu vực quan trọng. Kế hoạch này sẽ dựa trên sức mạnh tổng hợp của quân đội ta và kẻ thù hiện diện tại từng vị trí trong chiến trường. Việc lập kế hoạch cẩn thận này sẽ giúp tối ưu hóa nguồn lực và đảm bảo chiến thắng với tổn thất thấp nhất.

4.4.1 Mô tả nhiệm vụ

Quân đội ta cần xác định khả năng chiến thắng thông qua các bước phân tích và tính toán sau:

Chiến trường được mô phỏng bằng một **ma trận chiến sự** kích thước 10×10 . Mỗi ô trong ma trận đại diện cho một khu vực chiến sự, với giá trị số trong ô thể hiện sức mạnh của cứ điểm địch tại vị trí đó.

Để đạt được chiến thắng hoàn toàn, quân đội ta phải tiêu diệt được tất cả các cứ điểm của đối phương trên toàn bộ chiến trường, không để sót bất kỳ vị trí nào.

Để tiêu diệt mỗi cứ điểm của địch, quân ta phải sử dụng một phần sức mạnh tương ứng. Lượng sức mạnh cần thiết để tiêu diệt mỗi cứ điểm được tính theo công thức khác nhau tùy thuộc vào vị trí của cứ điểm (hàng chắn hay hàng lẻ trong ma trận).

Sức mạnh tổng hợp của quân đội ta được tính dựa trên ba yếu tố chính:

- Số lượng quân (**LF**): phản ánh quy mô lực lượng
- Kinh nghiệm chiến đấu (**EXP**): thể hiện khả năng chiến đấu
- Nguồn tiếp tế (**T**): đảm bảo khả năng duy trì chiến đấu

Quá trình tấn công được mô phỏng bằng cách duyệt qua từng ô trong ma trận chiến sự. Tại mỗi ô, sức mạnh của quân ta sẽ bị giảm đi một lượng tương ứng với sức mạnh của cứ điểm địch tại đó, theo công thức riêng cho từng loại vị trí.

Sau khi hoàn tất việc tấn công toàn bộ các cứ điểm, ta sẽ đánh giá kết quả dựa trên sức mạnh còn lại của quân đội ta:

- Nếu dương: Đủ sức mạnh để giành chiến thắng
- Nếu âm: Cần bổ sung thêm nguồn tiếp tế để đảm bảo chiến thắng

4.4.2 Mô tả hàm

- Tên hàm: `planAttack`
- Tham số đầu vào:
 - LF_1, LF_2 : Lực lượng quân đội tại quân đoàn 1 và 2.
 - EXP_1, EXP_2 : Mức độ kinh nghiệm của quân đoàn 1 và 2.
 - T_1, T_2 : Lượng tiếp tế hiện có tại quân đoàn 1 và 2.
 - `battleField[10][10]`: Ma trận biểu diễn sức mạnh địch tại mỗi vị trí.

- Quy trình xử lý:

- Tính sức mạnh tổng hợp ban đầu:

$$S = (LF_1 + LF_2) + (EXP_1 + EXP_2) \times 5 + (T_1 + T_2) \times 2$$

- Duyệt ma trận và cập nhật S tại mỗi vị trí (i, j) :

- * Hàng lẻ:

$$S = S - \left(\frac{\text{battleField}_{i,j} \times 3}{2} \right)$$

- * Hàng chẵn:

$$S = S - \left(\frac{\text{battleField}_{i,j} \times 2}{3} \right)$$

- Kết quả trả về:

- Giá trị sức mạnh tổng hợp S sau khi duyệt toàn bộ ma trận.

- Lưu ý:

- Không làm tròn lên giá trị của S trong quá trình duyệt ma trận.
- Nếu kết quả trả về là số thực, thì phải làm tròn lên.

- Giải thích:

- Giá trị dương: Sức mạnh của quân đội ta vượt trội, vì vậy, quân ta chắc chắn thắng.
- Giá trị âm: Số lượng tiếp tế cần bổ sung để có thể chiến thắng.

4.4.3 Ví dụ minh họa

Ví dụ 4.10

Trường hợp 1: Quân ta chiến thắng

Với dữ liệu nhập:

- $LF1 = 300, LF2 = 280$
- $EXP1 = 450, EXP2 = 470$
- $T1 = 2500, T2 = 2600$
- Ma trận battleField:

106	15	20	25	305	635	540	145	50	55
25	18	24	330	36	442	48	54	660	665
14	21	28	35	452	49	56	63	70	77
162	24	323	404	484	60	40	72	80	88
181	27	36	52	543	63	72	81	90	99
5	30	40	501	602	70	80	90	100	110
22	33	442	55	66	77	58	99	10	121
24	36	48	60	72	84	96	108	20	132
264	39	525	65	78	91	104	70	130	143
28	42	56	50	84	98	125	126	140	154

Tính toán sức mạnh ban đầu:

$$S = (300 + 280) + (450 + 470) \times 5 + (2500 + 2600) \times 2 = 15380$$

Sau khi hoàn tất quá trình tính toán trên ma trận:

$$S = 174$$

Vì $S > 0$, quân đội ta giành chiến thắng.

Hàm trả về 174.

Ví dụ 4.11

Trường hợp 2: Cần bổ sung tiếp tế

Với dữ liệu nhập:

- $LF1 = 300, LF2 = 280$
- $EXP1 = 450, EXP2 = 470$
- $T1 = 2500, T2 = 2600$
- Ma trận battleField:

506	15	20	25	305	635	540	145	50	55
325	18	24	330	36	442	48	54	660	665
14	21	28	35	452	49	56	63	70	77
562	24	323	404	484	60	40	72	80	88
181	27	36	52	543	63	72	81	90	99
505	30	40	501	602	70	80	90	100	110
922	33	442	55	66	77	58	99	110	121
124	36	48	60	72	84	96	108	120	132
264	39	525	65	78	91	104	170	130	143
28	42	56	50	84	98	125	126	140	154

Tính toán sức mạnh ban đầu:

$$S = (300 + 280) + (450 + 470) \times 5 + (2500 + 2600) \times 2 = 15380$$

Sau khi hoàn tất quá trình tính toán trên ma trận:

$$S = -2926$$

Vì $S < 0$, quân đội ta cần bổ sung 2926 đơn vị tiếp tế để có thể chiến thắng.

Hàm trả về -2926.

4.5 Nhiệm vụ 5: Bổ sung tiếp tế (2.0 điểm)

Sau khi thực hiện nhiệm vụ 4, nếu quân đội ta không đủ sức mạnh để chiến thắng, thì nhiệm vụ này sẽ tập trung vào việc lấy tiếp tế từ hậu phương để đảm bảo khả năng chiến đấu.

4.5.1 Mô tả nhiệm vụ

Nguồn tiếp tế bổ sung được lưu trữ trong một **ma trận 5×5** , trong đó mỗi ô chứa một lượng tiếp tế nhất định. Tổng lượng tiếp tế trong ma trận này không vượt quá 3000, và quân đội ta cũng không bị thiếu hụt quá 3000 đơn vị tiếp tế.

Theo diễn biến lịch sử, quân đội ta chắc chắn sẽ giành chiến thắng. Vì vậy, **trong mọi trường hợp**, ma trận tiếp tế sẽ luôn chứa ít nhất một tập hợp 5 ô có tổng giá trị lớn hơn lượng tiếp tế cần bổ sung.

Nhiệm vụ của quân đội ta là chọn đúng **5 gói tiếp tế** (tức 5 ô trong ma trận) sao cho:

- Tổng của 5 gói tiếp tế được chọn **lớn hơn hoặc bằng** lượng tiếp tế cần bổ sung (**shortfall**).
- Tổng lượng tiếp tế từ 5 ô được chọn phải là **nhỏ nhất có thể**, để tối ưu nguồn tiếp tế.

4.5.2 Mô tả hàm

- **Tên hàm:** resupply
- **Tham số đầu vào:**
 - **shortfall:** Lượng tiếp tế cần bổ sung.
 - **supply[5][5]:** Ma trận 5×5 chứa lượng tiếp tế tại mỗi vị trí.
- **Quy trình xử lý:**
 1. Duyệt qua ma trận **supply[5][5]** để xác định tất cả các ô có thể lấy tiếp tế.
 2. Chọn đúng **5 ô** sao cho tổng của chúng **lớn hơn hoặc bằng** lượng tiếp tế cần bổ sung (**shortfall**).
 3. Nếu có nhiều tổ hợp thỏa mãn, chọn tổ hợp có tổng tiếp tế **nhỏ nhất có thể**.
 4. Để tối ưu, có thể sắp xếp toàn bộ 25 ô theo giá trị tăng dần trước khi chọn tổ hợp phù hợp.
- **Precondition:** Ma trận tiếp tế luôn có ít nhất một tổ hợp hợp lệ.
- **Kết quả trả về:**
 - Tổng giá trị của 5 ô được chọn. Nếu có nhiều tổ hợp có cùng tổng giá trị nhỏ nhất, có thể trả về bất kỳ tổ hợp nào trong số đó.

4.5.3 Ví dụ minh họa

Ví dụ 4.12

Trường hợp 1: Tìm được tổ hợp tối ưu

Với dữ liệu nhập:

- $\text{shortfall} = 1050$
- Ma trận supply:

150	200	180	90	110
70	80	120	140	160
220	240	200	190	130
100	110	300	280	320
170	210	260	230	290

Chọn 5 ô tối ưu nhất: (70, 80, 280, 300, 320)

Tổng tiếp tế:

$$70 + 80 + 280 + 300 + 320 = 1050$$

Vì $1050 \geq 1050$ và đây là tổng nhỏ nhất có thể đáp ứng yêu cầu, nên đây là lựa chọn tối ưu.

Hàm trả về 1050.

4.6 Tạm kết

Sau những bước chuẩn bị kỹ lưỡng, quân đội ta đã hoàn thành tập kết lực lượng, sử dụng chiến thuật nghi binh để đánh lạc hướng kẻ địch, đảm bảo nguồn hậu cần ổn định và xây dựng kế hoạch tấn công chiến lược. Mỗi nhiệm vụ trong chiến dịch Tây Nguyên được thực hiện bài bản, thể hiện sự phối hợp nhịp nhàng giữa các quân đoàn và tinh thần quyết chiến, quyết thắng của quân đội ta.

Tuy nhiên, cuộc tổng tiến công chưa kết thúc. Sau khi giải phóng Tây Nguyên, cục diện chiến trường miền Nam thay đổi nhanh chóng. Địch hoảng loạn, rút lui khỏi nhiều vị trí trọng yếu, tạo thời cơ cho quân ta tiếp tục tiến công và giải phóng hoàn toàn miền Nam. Để đạt được mục tiêu cuối cùng, quân đội ta cần triển khai chiến dịch quyết định: Chiến dịch Hồ Chí Minh.

Liệu quân đội ta có thể tiếp tục duy trì sức mạnh, phá vỡ tuyến phòng thủ cuối cùng của địch và tiến vào Sài Gòn? Hãy cùng tiếp tục thực hiện những nhiệm vụ quan trọng trong Bài tập

lớn 2.

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

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

```
./main tnc_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 tay_nguyen_campaign.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 tay_nguyen_campaign.h, tay_nguyen_campaign.cpp để hoàn thành bài tập lớn này và đảm bảo hai yêu cầu sau:

- Chỉ có một lệnh **include** trong tập tin tay_nguyen_campaign.h là:

```
#include "main.h"
```

- Trong tập tin tay_nguyen_campaign.cpp, chỉ có một lệnh include:

#include "tay_nguyen_campaign.h"

- Ngoài hai include trên, không cho phép có bất kỳ lệnh **#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: `tay_nguyen_campaign.h` và `tay_nguyen_campaign.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

1. 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.
2. Mọi quyết định của giảng viên phụ trách bài tập lớn là quyết định cuối cùng.
3. Sinh viên không được cung cấp testcase sau khi chấm bài.
4. Nội dung Bài tập lớn sẽ được Harmony với một số câu hỏi trong bài Kiểm tra Cuối kỳ với nội dung tương tự.
5. Nếu sinh viên sử dụng mã nguồn từ các chương trình AI hoặc công cụ tự động tạo mã mà không hiểu rõ về cách hoạt động và ý nghĩa của mã đó, BTL sẽ bị chấm 0 điểm.

8 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.

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.**

9 Gian lận

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

- Có sự giống nhau bất thường giữa mã nguồn của các bài nộp. Trong trường hợp này, TẤT CẢ các bài nộp đều bị coi là gian lận. Do vậy sinh viên phải bảo vệ mã nguồn bài tập lớn của mình.
- 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.

10 Thay đổi so với phiên bản trước

1. Bỏ yêu cầu trả về "INVALID", khi chuỗi đầu vào có khoảng cách đối với phương pháp Caesar Cipher ở nhiệm vụ 2.2. Bổ sung điều kiện mã hóa đối với ký tự là số nguyên từ 0 đến 9.
2. Sửa lỗi Ví dụ 4.6 của nhiệm vụ 2.2.

3. Bổ sung công thức tính T1 và T2 với $E = 0$ ở nhiệm vụ 3.
4. Chỉnh sửa lại điều kiện để chọn 5 gói tiếp tế là "lớn hơn và bằng" ở nhiệm vụ 5.

Tài liệu

- [1] Phương Hà. Cuộc nghi binh lừa địch hoàn hảo trong Chiến dịch Tây Nguyên. *Báo Tin tức*, 2017. Truy cập ngày 07/02/2025.
- [2] Nguyễn Thành Hữu. Chiến dịch Tây Nguyên trong lịch sử Quân đội nhân dân Việt Nam. *Cổng thông tin điện tử Bộ Quốc phòng nước Cộng hòa Xã hội Chủ nghĩa Việt Nam*, 2013. Truy cập ngày 07/02/2025.
- [3] Đặng Cường. Ngày 4-3-1975: Quân đội nhân dân Việt Nam tiến hành Chiến dịch Tây Nguyên. *Báo Quân đội Nhân dân*, 2022. Truy cập ngày 07/02/2025.