Bài tập lớn số 2

Thiết kế chương trình phần mềm với các hàm và cấu trúc dữ liệu thích hợp để xử lý dữ liệu cảm biến.

I. Mô tả:

Sinh viên viết chương trình sử dụng ngôn ngữ C hoặc C++ đọc file có dữ liệu cảm biến được lưu trữ dưới dạng chuỗi số hex trong một file text có phần mở rộng là .dat sang dữ liệu text thông thường cho người dùng đọcvà lưu vào file csv và ngược lại. Tùy thuộc vào định dạng của file đầu vào chương trình phải tự động nhận dạng và xuất ra file đầu ra tương ứng. Chương trình phải chạy được từ command line với cấu trúc câu lệnh như sau:

```
C:\\ dust_convert [input_filename] [output_filename]
```

Trong đó `input_filename` là tên file đầu vào và `output_filename` là tên file đầu ra.
Ví dụ:

a. Câu lệnh

```
C:\\ dust_convert [data_filename.csv] [hex_filename.dat]
```

Sẽ chuyển đổi dữ liệu ở dạng text trong file `data_filename.csv` sang dạng hex trong `hex_filename.dat` giống như task 3 trong bài tập lớn số 1 (mini project 1).

b. Câu lệnh:

```
C:\\ dust_convert [hex_filename.dat] [data_filename.csv]
```

Sẽ chuyển đổi dữ liệu ở dạng hex trong `hex_filename.dat` sang dạng text trong file `data_filename.csv`.

II. Yêu cầu kỹ thuật:

1. Chuyển đổi dữ liệu:

Dữ liệu trong file .dat là bao gồm nhiều dòng, mỗi dòng thể hiện một chuỗi byte có cấu trúc như ở dưới đây

Start byte	Packet Length	ID	Time	PM2.5 concentration	AQI	Checksum	Stop byte
0x00 (1 byte)	1 byte	1 byte	4 bytes	4 bytes	2 byte	1 byte	0xFF (1 byte)

Trong đó ý nghĩa của các byte/nhóm byte như sau:

- Start byte (1 byte) là byte khởi đầu luôn có giá trị là 0x00.
- Stop byte (1 byte) là byte kết thúc luôn có giá trị là 0xFF.
- Packet length là độ dài của gói tin bao gồm cả start byte và stop byte.
- Id là số định danh của cảm biến luôn lớn hơn 0.
- Time là giá trị thời điểm đo theo định dạng thời gian trong hệ điều hành Unix được tính bằng giây.
- PM2.5 concentration là giá trị nồng độ bui, là một số thực 4 bytes (theo chuẩn IEEE 754).
- AQI là chỉ số chất lượng không khí, và là số nguyên 2 bytes
- Checksum là byte kiểm tra độ chính xác của dữ liệu trong gói tin được tính bằng mã bù 2 của các byte [packet length, id, time, PM2.5 concentration, AQI]

Dữ liệu trong file csv có dạng như ở ví dụ dưới, các trường dữ liệu trong mỗi dòng của file csv được ngăn cách bởi dấu “,”.

```
id,time,values,aqi,pollution
1,2022:11:26 00:01:00, 50.1,137,Slightly unhealthy
2,2022:11:26 00:01:00,24.2,76,Moderate
3,2022:11:26 00:01:00, 200.5,250,Very unhealthy
1,2022:11:26 00:02:00,100.2,174,Unhealthy
2,2022:11:26 00:02:00,10.4,43,Good
3,2022:11:26 00:02:00,160.9,210,Very unhealthy
…
```

Dòng đầu tiên là dòng tiêu đề bao gồm ***id, time, values, aqi, pollution***. Các dòng tiếp theo là các dữ liệu tương ứng được chuyển đổi từ các dòng dữ liệu ở file .dat, trong đó ***values*** là giá trị nồng độ bụi PM2.5, ***pollution*** là giá trị được xác định dựa trên bảng dưới:

Nồng độ c $[\mu g/m^3]$	$0 \leq c < 12$	$12 \leq c < 35.5$	$35.5 \leq c < 55.5$	$55.5 \leq c < 150.5$	$150.5 \leq c < 250.5$	$250.5 \leq c < 350.5$	$350.5 \leq c \leq 550.5$
AQI	0 ÷ < 50	50 ÷ <100	100 ÷ <150	150 ÷ <200	200 ÷ <300	300 ÷ <400	400 ÷ 500
Cấp độ ô nhiễm	Good	Moderate	Slightly unhealthy	Unhealthy	Very unhealthy	Hazardous	Extremely hazardous

Ví dụ:

Dòng "1,2022:11:26 00:01:00, 50.1,137,Slightly unhealthy" trong file `data_filename.csv` sẽ được chuyển đổi thành "00 0F 01 63 81 57 3C 42 48 66 66 00 89 9A FF" là 1 dòng trong file `hex_filename.dat` nếu gọi câu lệnh trong ví dụ a. ở mục I và ngược lại nếu gọi câu lệnh ví dụ b mục I.

2. Sắp xếp dữ liệu

Trong trường hợp chuyển đổi từ kiểu dữ liệu dạng hex sang dạng text, viết chức năng cho phép người dùng chuyển đổi và sắp xếp dữ liệu bằng cách thêm các tham số của câu lệnh command line như dưới đây.

```
C:\\ dust_convert [hex_filename.dat] [data_filename.csv] -s [params] [-asc/-dsc]
```

Trong đó:

- `-s` : để xác định câu lệnh sẽ thực hiện việc sắp xếp dữ liệu
- `[params]` : là danh sách các tham số sử dụng để sắp xếp, thứ tự các tham số từ trái qua phải biểu diễn thứ tự ưu tiên khi sắp xếp. Có 3 tham số có thể sử dụng là id, time và values; các tham số được viết cách nhau bởi dấu phẩy nếu sử dụng nhiều hơn 1 tham số. Nếu không cung cấp `[params]` thì thứ tự ưu tiêu mặc định khi sắp xếp sẽ là id, time và values.
- `[-asc/-dsc]` : là tham số có giá trị là -asc hoặc -dsc để xác định việc sắp xếp từ nhỏ đến lớn hay từ lớn đến nhỏ

Ví dụ: câu lệnh

```
C:\\ dust_convert [hex_filename.dat] [data_filename.csv] -s time,id -asc
```

Sẽ chuyển đổi dữ liệu từ file .dat sang file .csv và sắp xếp dữ liệu trong file .csv theo thời gian (time) tăng dần và 2 dòng có giá trị thời gian bằng nhau thì dòng nào có id nhỏ hơn sẽ xếp trước.

Chú ý: sinh viên cần lựa chọn ít nhất 2 thuật toán sắp xếp khác nhau và ghi lại thời gian thực hiện mỗi thuật toán. Sinh viên phải tự viết lại code thuật toán sắp xếp, không dùng hàm trong thư viện. Trong các thuật toán được lựa chọn, sinh viên được khuyến khích tìm hiểu thêm và sử dụng ít nhất một thuật toán không có trong bài giảng.

3. Thông số chạy chương trình

Chương trình chạy với command-line cần lưu lại các thông số chạy chương trình vào 1 log file có tên là **dust_convert_run.log**. Các thông tin cần lưu lại như sau (trong phần đóng khung):

Dòng số	
1	Total number of rows: 1000
2	Succesfully converted rows: 990
3	Error rows: 10
4	Sorting algorithm bubble [ms]: 0.06
5	Sorting algorithm insertion [ms]: 1
6	Sorting algorithm quick [ms]: 0.01

- Total number of rows là số dòng trong file đầu vào, ở ví dụ trong bảng trên thì file đầu vào có 1000 dòng dữ liệu. Các dòng trống (không có ký tự nào mà chỉ có ký tự dấu cách và/hoặc ký tự xuống xòng "\n") thì bỏ qua không tính.
- Succesfully converted rows số dòng chứa dữ liệu hợp lệ và được chuyển đổi thành công
- Error rows số dòng chữa dữ liệu lỗi.
- Các dòng tiếp theo là thời gian thực hiện thuật toán sắp xếp (nếu có), đơn vị là ms với định dạng:
  Sorting algorithm NAME_OF_THE_ALGORITHM [ms]: x
  Với NAME_OF_THE_ALGORITHM là tên thuật toán được áp dụng, x là thời gian thực hiện tính bằng ms.

4. Xử lý lỗi:

Chương trình chạy với command-line cần lưu lại các lỗi xảy ra vào 1 log file có tên là **dust_convert_error.log**. Mỗi một lỗi chương trình có thông báo lỗi được ghi trên một dòng của log file với định dạng như sau:

Error AB: DESCRIPTION

Trong đó

- AB là mã lỗi, là một số nguyên có 2 chữ số (nếu số < 10 thì sẽ ghi thêm số 0 phía trước, ví dụ 01, 02, …)
- DESCRIPTION là mô tả lỗi (khuyến khích ghi bằng tiếng Anh, không sử dụng tiếng việt có dấu).

a. *Lỗi chung:*

- Sai câu lệnh command-line ví dụ thiếu 1 hoặc nhiều tham số. Thông báo lỗi có thể là: "Error 01: invalid command"
- File đầu vào hoặc đầu ra đã tồn tại nhưng không cho phép truy cập. Thông báo lỗi có thể là "Error 02: denied access FILENAME" trong đó FILENAME được thay bằng tên của file không truy cập được.
- File đầu vào có nội dung không phải theo định dạng csv hoặc hex như đã quy định. Thông báo lỗi "Error 03: invalid file format"

b. *Một số lỗi có thể gặp ở khi chuyển từ file csv sang file hex:*

- Lỗi dữ liệu trong file csv.
  - Tất cả các trường dữ liệu trên một dòng đều bị bỏ trống, ví dụ: ", , "
  - Id bị bỏ trống hoặc không hợp lệ, ví dụ "-1,2022:11:26 00:00:00, 50.1"
  - Thời gian bị bỏ trống hoặc không hợp lệ, ví dụ "1,2022:11:26 00:00:, 50.1"
  - Giá trị nồng độ bụi bị bỏ trống, ví dụ "1,2022:11:26 00:00:00, "

  Đối với lỗi dự liệu, thông báo lỗi phải chỉ rõ lỗi ở dòng nào với cấu trúc thông báo lỗi như sau: "Error 04: data is missing at line X" trong đó X là số thự tự dòng trong file đầu vào với dòng tiêu đề (tên các trường dữ liệu) "id,time,values" được coi là dòng số 0, dòng tiếp theo sau dòng tiêu đề là 1.

- Dữ liệu trùng lặp trong file hex tức là 2 hoặc nhiều dòng có dữ liệu giống hệt nhau, có thể thông báo lỗi như sau: "Error 05: duplicated data at lines X1, X2" trong đó X1, X2 là số thự tự dòng trong file đầu vào tương tự như trên.

Trong trường hợp file csv có lỗi sai/thiếu dữ liệu (missing data) hoặc trùng lặp dữ liệu (duplicated data), việc chuyển đổi vẫn phải thực hiện bình thường với các dòng có dữ liệu hợp lệ khác, dòng có dữ liệu lỗi sẽ được bỏ qua, các dòng có dữ liệu trùng lặp thì chỉ cần chuyển đổi 1 lần.

c. Một số lỗi có thể gặp ở khi chuyển từ file hex sang file csv:

- Lỗi dữ liệu trong file hex.
    - Sai giá trị Start byte (1 byte) và Stop byte (1 byte).
    - Packet length sai.
    - Time sai (thời gian ở tương lai, so với thời điểm đọc file đầu vào).
    - PM2.5 concentration là giá trị nồng độ bụi và AQI không nhất quán với nhau.
    - Checksum sai.

  Đối với lỗi dự liệu, thông báo lỗi phải chỉ rõ lỗi ở dòng nào với cấu trúc thông báo lỗi như sau: "Error 06: invalid data packet at line X" trong đó X là số thự tự dòng trong file đầu vào.
- Dữ liệu trùng lặp trong file csv tức là 2 hoặc nhiều dòng có dữ liệu giống hệt nhau, có thể thông báo lỗi như sau: "Error 05: duplicated data at lines X1, X2" trong đó X1, X2 là số thứ tự dòng trong file đầu vào tương tự như trên.

Trong trường hợp file hex có lỗi sai/thiếu dữ liệu (invalid data packet) hoặc trùng lặp dữ liệu (duplicated data), việc chuyển đổi vẫn phải thực hiện bình thường với các dòng có dữ liệu hợp lệ khác, dòng có dữ liệu lỗi sẽ được bỏ qua, các dòng có dữ liệu trùng lặp thì chỉ cần chuyển đổi 1 lần.

d. Các lỗi khác:

Sinh viên có thể tự đề xuất thêm các lỗi khác, tuy nhiên cần phải mô tả các lỗi khác đó trong báo cáo.

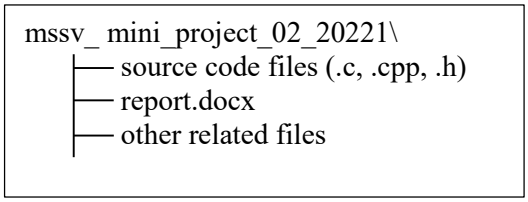
III. Thiết kế chương trình:

Sinh viên sử dụng các tiếp cận từ trên xuống (top-down approach) để thiết kế chương trình. Do đó, yêu cầu sinh viên phải vẽ sơ đồ top-down approach để minh họa cách phân hoạch hàm và quan hệ giữa các hàm trong chương trình trong báo cáo kèm theo mô tả/giải thích ngắn gọn.

Sinh viên cũng phải vẽ ít nhất 2 lưu đồ thuật toán:

- 1 lưu đồ thuật toán tổng quát cho toàn bộ chương trình
- Và 1 lưu đồ thuật toán cho 1 hàm quan trọng trong các hàm đã thiết kế (tùy chọn có thể vẽ nhiều hơn 1).

Sinh viên cũng phải vẽ sơ đồ cấu trúc thư mục và các file liên quan trong bài tập lớn theo định dạng như sau trong báo cáo cáo kèm theo mô tả/giải thích ngắn gọn các files trong folder đặc biệt là các file source code nếu source code được chia thành nhiều files.



Trong đó thư mục chứa bài tập có tên là "mssv_mini_ project_02_20221" với "mssv" thay bằng mã số sinh viên. Các file source code, "report.docx" (là file báo cáo) và các file liên quan khác nằm trong cùng thư mục này, không sử dụng thư mục con nào khác.

IV. Coding styles (phong cách lập trình):

Coding style cần nhất quán trong toàn bộ chương trình và tuân theo quy định GNU mô tả trong link sau: https://www.gnu.org/prep/standards/html_node/Writing-C.html

Một cách ngắn gọn:

- Mã chương trình cần được trình bày gọn gàng, dễ theo dõi bằng cách lùi dòng, sử dụng dấu {}, ngắt dòng và cách dòng hợp lý.

- Cung cấp chú thích (comment) trong code rõ ràng, dễ hiểu để giải thích rõ hơn chương trình.
- Tên hàm và tên biến nên được đặt theo tiếng Anh, ngắn gọn và có tính tự mô tả.
- Tránh "hard-coding".

Lưu ý: Sinh viên cũng không được sử dụng các thư viện khác ngoài các thư viện chuẩn của C/C++.

V. Công cụ lập trình:

Editor: Visual studio code (https://code.visualstudio.com/download)
Compiler: gcc or g++ in MinGW-w64 (https://sourceforge.net/projects/mingw-w64/)
Sinh viên cũng cần mô tả rõ chương trình viết trong hệ điều hành nào (Windows, Linux, MacOS) trong báo cáo.

VI. Báo cáo và hướng dẫn nộp:

- Sinh viên làm bài tập theo cá nhân hoặc theo nhóm nếu phân theo nhóm.
- Toàn bộ bài tập lớn phải được tổ chức trong một thư mục như mô tả trong mục III.
- Sinh viên viết báo cáo là file **word** không quá 6 trang A4 (không bao gồm code chương trình và không nên đưa code vào báo cáo) sử dụng IEEE template (có đính kèm trong Team Assignment). Trong thông tin tác giả dưới tiêu đề cần ghi rõ họ và tên, mã số sinh viên.
- Nội dung chính của báo cáo bao gồm:
    - Ý tưởng chính: mô tả ý tưởng thiết kế chương trình, bao gồm sơ đồ top-down approach, cấu trúc thư mục, các file code (nếu phân chia thành nhiều file code) và các thư viện được sử dụng.
    - Thiết kế chi tiết: mô tả thiết kế các hàm (nếu quá nhiều hàm thì cần mô tả chi tiết các hàm quan trọng nhất) bao gồm khuôn mẫu hàm (tên, kiểu trả về, danh sách tham biến) và mô tả đầu vào/ra. Chọn ít nhất 2 hàm quan trọng (trong đó 1 là luồng chương trình chính) và vẽ lưu đồ thuật toán.
    - Kết quả và đánh giá: mô tả kết quả chạy chương trình và đánh giá chất lượng chương trình
    - Kết luận: nêu vắn tắt lại những vấn đề đã thực hiện được và chưa thực hiện được.
    - Tài liệu tham khảo (nếu có).
- Sinh viên nén toàn bộ thư mục "mssv_mini_project_02_20221" thành file "mssv_mini_project_02_20221.zip" (chú ý là file .zip không sử dụng file .rar hay bất kỳ định dạng file nén nào khác).
    - Trong thư mục nộp bài chỉ giữ lại các file code, file chạy chương trình và file báo cáo bằng định dạng word (file docx).
    - Tất cả các file không liên quan đến bài tập lớn cần phải xóa trước khi nén và nộp.
    - mssv trong tên thư mục và tên file nén thay bằng mã số sinh viên

    Ví dụ: sinh viên Nguyễn Văn A có mã số sinh viên là 20221234 nộp bài "20221234_ mini_project_02_20221.zip"
- Bài làm phải được nộp qua Team Assignment đúng hạn, không nộp bài qua email hay bất cứ kênh nào khác.

VII. Đánh giá:

- Bài tập lớn sẽ được chấm như sau:
    - Hoàn thành tất cả các tasks, chương trình chạy không có lỗi gì, xử lý tốt các trường hợp lỗi và có sáng tạo (4 điểm).
    - Thiết kế tốt và có tính sử dụng lại cao (2 điểm).
    - Phong cách lập trình tốt (good coding style) (2 điểm).
    - Báo cáo trình bày đúng template, bố cục rõ ràng, trình bày dễ hiểu, không có lỗi chính tả ngữ pháp (2 điểm).
    - Điểm thưởng nếu sinh viên thực hiện được ít nhất 2 thuật toán sắp xếp khác không phải là các thuật toán đã được giới thiệu trong bài giảng (2-3 điểm)
- Sinh viên phải tự thực hiện bài tập lớn. Không copy bài của bạn khác và nên giữ bí mật bài làm của mình. **Nếu hai hay nhiều sinh viên có mã nguồn và/hoặc báo cáo giống nhau dù chỉ một**

phần thì bài làm của tất cả các sinh viên liên quan sẽ bị coi là pham quy và coi như không nộp bài không cần biết ai copy bài của ai.

- Sinh viên chú ý nộp đúng hạn trên Team Assignment. Không được phép nộp muộn.
- Ngoài bản mềm nộp trên Team sinh viên cần in báo cáo bản cứng (khổ A4, in 2 mặt) và nộp khi đi thi cuối kỳ.
- Sinh viên không nộp bài sẽ không có cơ sở để hỏi thi vấn đáp và sẽ nhận điểm cuối kỳ <3.
- Điểm thi cuối kỳ sẽ được đánh giá tổng hợp giữa báo cáo và phần vấn đáp.

---------- Hết ----------