

## Bài tập lớn

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++ với các hàm và cấu trúc dữ liệu thích hợp để mô phỏng dữ liệu cảm biến bụi PM2.5 đo nồng độ hạt bụi có kích thước  $< 2.5$  microns trong không khí.

- Phạm vi đo:  $0 \div 800 \mu g/m^3$
- Độ phân giải:  $0.1 \mu g/m^3$

Các nhiệm vụ yêu cầu bao gồm:

#### 1. Task 1:

Viết chương trình cho phép người dùng cung cấp số lượng cảm biến, tần số trích mẫu và khoảng thời gian đo bằng cách gõ lệnh trên command-line với cấu trúc câu lệnh như sau:

```
C:\dust_sim -n [num_sensors] -st [sampling] -si [interval]
```

Trong đó:

- `dust_sim`: là tên file chương trình đã biên dịch
- `-n [num_sensors]` là cặp tham số đầu vào để cung cấp số lượng cảm biến, `[num_sensors]` cần được thay thế bởi một số cụ thể. Chương trình cần đưa ra thông báo lỗi nếu chỉ một trong 2 tham số này xuất hiện. Nếu cả 2 thông số này không xuất hiện trong câu lệnh command-line thì chương trình sẽ lấy số lượng cảm biến mặc định là **1 (một)**.
- `-st [sampling]` là cặp tham số để cung cấp thời gian trích mẫu với `[sampling]` cần được thay thế bởi một số nguyên dương với đơn vị là **giây**, thời gian trích mẫu nhỏ nhất cho phép là **10 giây**. Chương trình cần đưa ra thông báo lỗi nếu chỉ một trong 2 tham số này xuất hiện. Nếu cả 2 thông số này không xuất hiện trong câu lệnh command-line thì chương trình sẽ lấy tần số trích mẫu mặc định là **60 giây**.
- `-si [interval]` là cặp tham số để cung cấp khoảng thời gian đo với `[interval]` cần được thay thế bởi một số nguyên dương đơn vị là **giờ**, khoảng thời gian mô phỏng nhỏ nhất là **1 giờ**. Chương trình cần đưa ra thông báo lỗi nếu chỉ một trong 2 tham số này xuất hiện. Nếu cả 2 thông số này không xuất hiện trong câu lệnh command-line thì chương trình sẽ lấy tần số trích mẫu mặc định là **24 giờ**.

Chương trình sẽ xuất ra một tập dữ liệu bao gồm định danh của cảm biến (sensor id), thời điểm đo (timestamp) mô phỏng và giá trị cảm biến mô phỏng (values), với thời điểm bắt đầu mô phỏng là thời điểm hiện tại lấy từ giờ của hệ thống (giờ trong máy tính) trừ đi khoảng khoảng thời gian mô phỏng.

- Số định danh (id) của cảm biến là các số từ 1 đến `num_sensors` với `num_sensors` là số lượng cảm biến mà người dùng cung cấp trong câu lệnh command-line, ví dụ `num_sensors = 10` thì chương trình sẽ tạo ra 10 cảm biến có id là 1, 2, 3, ..., 10.
- Thời điểm đo (mô phỏng) có định dạng là `YYYY:MM:DD hh:mm:ss`, trong đó
  - `YY` – năm, `MM` – tháng, `DD` – ngày.
  - `hh` – giờ, `mm` – phút, `ss` – giây.Ví dụ: `2024:11:01 08:30:02`
- Giá trị đo mô phỏng được tạo ra ngẫu nhiên là một số thực, với độ chính xác là 1 chữ số sau dấu phẩy.

**Chú ý:** thời gian mô phỏng không phải là thời gian thực mà là thời gian mô phỏng tính toán do đó, sinh viên không dùng hàm tạo trễ như `sleep()` hoặc các vòng lặp tạo trễ.

Dữ liệu mô phỏng xuất ra sẽ được lưu vào một file có tên là “`dust_sensor.csv`”, nếu file đã tồn tại thì ghi đè lên file cũ. File này tuân theo định dạng CSV (comma-separated values), tức là các trường dữ

liệu cách nhau bởi dấu phẩy. Sinh viên có thể tham khảo thêm về định dạng csv ở link sau: <https://www.ietf.org/rfc/rfc4180.txt>

Ví dụ câu lệnh command-line: `C:\dust_sim -n 3 -st 60 -si 10`

- Giả sử giờ hệ thống tại thời điểm chạy câu lệnh command-line là 2024:11:02 10:00:00, thời điểm bắt đầu mô phỏng sẽ là 2024:11:02 00:00:00. Thời gian mô phỏng bao gồm cả thời điểm bắt đầu và thời điểm chạy câu lệnh.
- Dữ liệu trong file “dust\_sensor.csv” sẽ có dạng như sau:

```
id,time,value
1,2024:11:02 00:00:00, 50.1
2,2024:11:02 00:00:00,24.2
3,2024:11:02 00:00:00, 200.5
1,2024:11:02 00:01:00,100.2
2,2024:11:02 00:01:00,55.4
3,2024:11:02 00:01:00,160.9
...
1,2024:11:02 10:00:00,120.2
2,2024:11:02 10:00:00,90.4
3,2024:11:02 10:00:00,351.0
```

Trong đó dòng đầu tiên “id,time,values” là dòng tiêu đề của các trường dữ liệu.

## 2. Task 2:

Viết một chương trình xử lý dữ liệu trong một file csv có định dạng như ở task 1. Chương trình phải được chạy bằng câu lệnh command-line như dưới đây.

`C:\dust_process [data_filename.csv]`

Trong đó:

- `dust_process`: là file chương trình đã biên dịch
- `[data_filename.csv]` là file csv chứa dữ liệu cảm biến bụi. Nếu người dùng không cung cấp tên file thì mà chỉ gõ `C:\dust_process` thì chương trình sẽ sử dụng tên file mặc định là “dust\_sensor.csv”.

Ví dụ: `C:\dust_process dust_sensor_ee3491.csv`

Chương trình phải có khả năng xử lý được tối đa 10000 điểm dữ liệu, tức là file đầu vào `data_filename.csv` có thể chứa ít nhất 10000 dòng dữ liệu.

### a. Task 2.1:

Giả sử môi trường cần đo có nồng độ bụi dao động trong khoảng từ  $3 \div 550.5 \mu g/m^3$ , chương trình cần thực hiện việc kiểm tra dữ liệu hợp lệ trong file csv. Các giá trị đo nằm ngoài khoảng trên đều là giá trị dị biệt và cần loại bỏ. Các điểm dữ liệu dị biệt này cần phải được lưu trong một file csv có tên là “dust\_outlier.csv”, có định dạng như sau:

```
number of outliers: 3
id,time,value
1,2024:11:02 00:00:00,2.1
3,2024:11:02 19:03:00,-1.0
3,2024:11:02 21:06:00,560.2
```

Trong đó dòng đầu tiên được “number of outliers: X” với X là số lượng các điểm dữ liệu lọc ra từ file ban đầu, trong ví dụ trên thì  $X = 3$ . Nếu không có giá trị dị biệt nào thì  $X = 0$  và không có dòng dữ liệu nào trong file này.

Các giá trị hợp lệ được lưu vào 1 file csv khác đặt tên là “dust\_valid.csv” có định dạng giống với file gốc.

**Sử dụng các giá trị đo hợp lệ để thực hiện các phần tiếp theo từ task 2.2 đến hết.**

b. Tasks 2.2:

Nồng độ bụi có thể được quy đổi về chỉ số chất lượng không khí (AQI – Air quality index) như sau:

*Bảng 1. Chỉ số chất lượng không khí và cấp độ ô nhiễm*

Nồng độ bụi $c$ [ $\mu g/m^3$ ]	AQI	Cấp độ ô nhiễm	Mã ô nhiễm
$0 \leq c < 12$	$0 \div < 50$	Good	A
$12 \leq c < 35.5$	$50 \div < 100$	Moderate	B
$35.5 \leq c < 55.5$	$100 \div < 150$	Slightly unhealthy	C
$55.5 \leq c < 150.5$	$150 \div < 200$	Unhealthy	D
$150.5 \leq c < 250.5$	$200 \div < 300$	Very unhealthy	E
$250.5 \leq c < 350.5$	$300 \div < 400$	Hazardous	F
$350.5 \leq c \leq 550.5$	$400 \div 500$	Extremely hazardous	G

Chương trình cần tính nồng độ bụi trung bình theo từng giờ, ví dụ giá trị trung bình tại thời điểm 2022:11:26 02:00:00 là nồng độ bụi trung bình trong khoảng từ 2022:11:26 01:00:00 đến 2022:11:26 01:59:59 của cảm biến. Sau đó chương trình cần xác định chỉ số AQI tương ứng với giá trị trung bình đó và cấp độ ô nhiễm. Lưu kết quả tính toán ra một file csv có tên là “dust\_aqi.csv” có định dạng như sau:

```
id,time,value,aqi,pollution
1,2024:10:28 23:00:00,8.5,35,Good
2,2024:10:28 23:00:00,64.8,155,Unhealthy
3,2024:10:28 23:00:00,197.2,247,Very unhealthy
4,2024:10:28 23:00:00,239.7,289,Very unhealthy
5,2024:10:28 23:00:00,134.9,192,Unhealthy
1,2024:10:29 00:00:00,42.9,118,Slightly unhealthy
2,2024:10:29 00:00:00,89.7,168,Unhealthy
3,2024:10:29 00:00:00,249.4,299,Very unhealthy
...
```

c. Tasks 2.3:

Xác định giá trị lớn nhất (max), nhỏ (min), giá trị nồng độ bụi trung bình (mean) và số trung vị (median) của các giá trị nồng độ bụi đo được tại mỗi cảm biến và lưu kết quả vào file có tên là “dust\_summary.csv” có định dạng như ở dưới.

```
id,parameter,time,value
1,max,2024:10:29 14:17:39,529.0
1,min,2024:10:29 19:01:39,3.1
1,mean,10:00:00,71.03
1,median,10:00:00,46.5
2,max,2024:10:29 21:47:39,546.2
2,min,2024:10:29 12:59:39,3.0
2,mean,10:00:00,36.36
2,median,10:00:00,8.9
3,max,2024:10:29 21:41:39,508.1
3,min,2024:10:29 21:51:39,3.6
3,mean,10:00:00,88.51
3,median,10:00:00,48.2
...
```

Thời điểm tương ứng với các giá trị lớn nhất và nhỏ nhất trong file trên là thời điểm sớm nhất mà các giá trị này xuất hiện trong file đầu vào. Thời gian tương ứng với giá trị mean và trung vị (median) là khoảng thời gian mô phỏng.

d. Task 2.4:

Dựa trên dữ liệu tính toán ở task 2.2, thống kê tổng số giờ ở mỗi cấp độ ô nhiễm đo được tại các node cảm biến và lưu vào một file có tên là “dust\_statistics.csv”.

```
id, pollution,duration
1,Good,2
1, Moderate,1
1, Slightly unhealthy,3
1,Unhealthy,0
1,Very unhealthy,2
1, Hazardous,2
1, Extremely hazardous,0
2,Good,1
2, Moderate,2
2, Slightly unhealthy,0
2,Unhealthy,4
2,Very unhealthy,1
2, Hazardous,0
2, Extremely hazardous,1
...
```

3. Task 3:

Giả sử ta cần gửi dữ liệu thu được ở task 2.2 qua một giao thức truyền thông bằng cách tạo lập một bản tin truyền thông là một chuỗi byte có cấu trúc như ở dưới đây

Start byte	Packet Length	ID	Time	PM2.5	AQI	Pollution code	Checksum	Stop byte
0xAA (1 byte)	1 byte	1 byte	4 bytes	4 bytes	2 byte	1 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 0xAA.
- 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 bụi, là một số thực 4 bytes (theo chuẩn IEEE 754).
- AQI là chỉ số chất lượng không khí, và là số nguyên 2 bytes.
- Pollution code là mã ASCII tương ứng với ký tự biểu diễn mã ô nhiễm của giá trị AQI ở bảng 1.
- 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, pollution code]

Tất cả các giá trị số (số nguyên và số thực) đều được biểu diễn bằng **little-endian**.

Khi người dùng viết câu lệnh command-line như dưới đây

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

thì chương trình sẽ:

- đọc từng dòng của file đầu vào,
- chuyển đổi dữ liệu sang dạng gói tin là một chuỗi byte như mô tả ở trên, mỗi byte cách nhau bởi dấu cách, các byte được ghi dưới dạng số hex.
- và ghi mỗi gói tin này trên một dòng trong file đầu ra tương ứng.

- nếu file `hex_filename.dat` đã tồn tại thì sẽ ghi đè lên file cũ.

Trong câu lệnh trên:

- `data_filename.csv` là file đầu vào đuôi CSV có định dạng như ở task 2.2
- `hex_filename.dat` là file đầu ra có dạng text, với phần mở rộng là `dat`

Ví dụ:

`C:\dust_convert dust_aqi.csv hex_packet_ee3491.dat`

Dòng “2,2024:09:18 20:06:11,499.7,475,Extremely hazardous” trong file `dust_aqi.csv` sẽ được chuyển đổi thành:

`AA 10 02 43 D0 EA 66 9A D9 F9 43 DB 01 47 B9 FF`

## II. Yêu cầu kỹ thuật khác:

Thời gian chạy của mỗi task (1, 2, 3) không quá 30 giây.

Chương trình chạy với command-line cần lưu lại các lỗi xảy ra vào 1 log file, với mỗi task ở mục I sẽ phải có 1 log file tương ứng đặt tên là `task1.log`, `task2.log` và `task3.log`. Mỗi một lỗi 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).

### 1. Một số lỗi có thể gặp ở task 1:

- Sai câu lệnh command-line ví dụ thiếu 1 hoặc nhiều tham số. Thông báo lỗi có thể là: “Error 01: invalid command”
- Sai định dạng các thông số đầu vào, ví dụ số lượng cảm biến < 0. Thông báo lỗi có thể là “Error 02: invalid argument”
- File “`dust_sensor.csv`” đã tồn tại nhưng không cho phép ghi đè. Thông báo lỗi có thể là “Error 03: denied access dust\_sensor.csv”

### 2. Một số lỗi có thể gặp ở task 2:

- File đầu vào `data_filename.csv` không tồn tại hoặc không cho phép truy cập. Thông báo lỗi: “Error 01: file not found or cannot be accessed”
- File đầu vào `data_filename.csv` có nội dung không phải theo định dạng csv như đã quy định. Thông báo lỗi “Error 02: invalid csv file”
- Người dùng gõ sai định dạng câu lệnh command-line. Thông báo lỗi “Error 03: invalid command”
- Lỗi dữ liệu trong file csv.
  - o Tất cả các trường dữ liệu trên một dòng đều bị bỏ trống, ví dụ: “, , ”
  - o Id bị bỏ trống hoặc không hợp lệ, ví dụ “-1,2022:11:26 00:00:00, 50.1”
  - o Thời gian bị bỏ trống hoặc không hợp lệ, ví dụ “1,2022:11:26 00:00:, 50.1”
  - o Giá trị nồng độ bụi bị bỏ trống, ví dụ “1,2022:11:26 00:00:00, ”

Đối với lỗi dữ 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 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.

Chương trình cần bỏ qua dòng lỗi dữ liệu này và tiếp tục xử lý các dòng tiếp theo khi thực hiện task 2.

- File đầu vào chứa dữ liệu bị lặp:
  - o File đầu vào chứa 2 hoặc nhiều dòng giống nhau ít nhất cả 2 trường thông tin id và thời gian.

Thông báo lỗi cần chỉ rõ dòng nào trong trùng dữ liệu với dòng nào trong file đầu vào, ví dụ: “Error 05: data is duplicated at line X and Y” với X và Y là số thứ tự của 2 dòng có dữ liệu trùng nhau. 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.

Chương trình cần tiếp tục xử lý các dòng dữ liệu không trùng cũng như chỉ dòng dữ liệu đầu tiên trong các dòng dữ liệu trùng nhau khi thực hiện task 2 và bỏ qua các dòng có dữ liệu trùng còn lại.

- Nếu file đầu vào có nhiều hơn 10000 dòng dữ liệu, chương trình cần xử lý 10000 dòng dữ liệu đầu tiên và bỏ qua các dòng còn lại. Thông báo lỗi cần có là: “Error 06: input file is too large”

### 3. Một số lỗi có thể gặp ở task 3:

- File đầu vào `data_filename.csv` không tồn tại hoặc không cho phép truy cập. Thông báo lỗi: “Error 01: file not found or cannot be accessed”
- File đầu vào `data_filename.csv` có nội dung không phải theo định dạng csv như đã quy định. Thông báo lỗi “Error 02: invalid csv file”
- Người dùng gõ sai định dạng câu lệnh command-line. Thông báo lỗi “Error 03: invalid command”
- Lỗi dữ liệu trong file csv.

- o Tất cả các trường dữ liệu trên một dòng đều bị bỏ trống, ví dụ: “, , , ,”
- o Id bị bỏ trống hoặc không hợp lệ, ví dụ “-1,2024:09:18 20:06:11,11.1,46,Good”
- o Thời gian bị bỏ trống hoặc không hợp lệ, ví dụ “1,2024:09:18 20:06:71,11.1,46,Good”
- o Giá trị nồng độ bụi và/hoặc AQI bị bỏ trống, ví dụ “1,2024:09:18 20:06:11, ,46,Good”
- o AQI bị bỏ trống hoặc không hợp lệ, ví dụ: “1,2024:09:18 20:06:11,11.1, ,Good”
- o Giá trị nồng độ bụi và AQI không nhất quán với nhau như mô tả trong bảng 1:

Đối với lỗi dữ 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 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.

- File đầu vào chứa dữ liệu bị lặp:
  - o File đầu vào chứa 2 hoặc nhiều dòng giống nhau ít nhất cả 2 trường thông tin id và thời gian.

Thông báo lỗi cần chỉ rõ dòng nào trong trùng dữ liệu với dòng nào trong file đầu vào, ví dụ: “Error 05: data is duplicated at line X and Y” với X và Y là số thứ tự của 2 dòng có dữ liệu trùng nhau. 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.

Chương trình cần tiếp tục xử lý các dòng dữ liệu không trùng cũng như chỉ dòng dữ liệu đầu tiên trong các dòng dữ liệu trùng nhau khi thực hiện task 2 và bỏ qua các dòng có dữ liệu trùng còn lại.

- Nếu file đầu vào có nhiều hơn 10000 dòng dữ liệu, chương trình cần xử lý 10000 dòng dữ liệu đầu tiên và bỏ qua các dòng còn lại. Thông báo lỗi cần có là: “Error 06: input file is too large”
- Nếu file đầu ra, ví dụ: `hex_filename.dat`, đã tồn tại và không thể bị ghi đè. Thông báo lỗi cần có là: “Error 07: cannot override the hex file”

### 4. Các trường hợp 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:

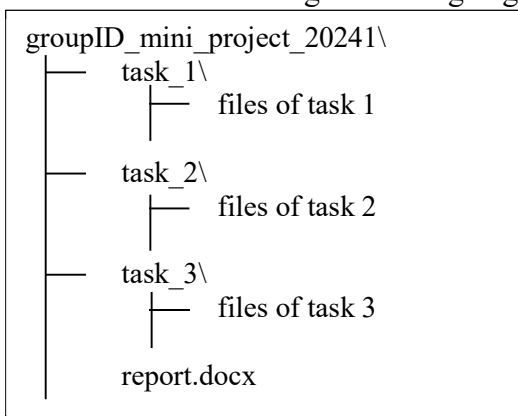
Sinh viên cần trình bày phân tích yêu cầu và ý tưởng thiết kế (các) chương trình một cách rõ ràng và hợp lý.

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 các lưu đồ thuật toán:

- 1 lưu đồ thuật toán tổng quan cho toàn bộ chương trình của mỗi task.
- Và ít nhất 1 lưu đồ thuật toán cho 1 hàm quan trọng trong các hàm đã thiết kế trong mỗi task (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 kèm theo mô tả/giải thích ngắn gọn.:



Trong đó thư mục gốc có tên là “mssv\_mini\_project\_20241” và 3 thư mục con “task\_1”, “task\_2” và “task\_3” chứa các file liên quan. “report.docx” là file báo cáo và nằm trong thư mục gốc, “groupID” thay bằng mã nhóm. 3 thư mục con “task\_1”, “task\_2” và “task\_3” không nên chứa bất kỳ thư mục con nào.

Sinh viên cũng cần trình bày cấu trúc dữ liệu quan trọng đã được sử dụng hoặc định nghĩa trong bài tập lớn và giải thích.

#### 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](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 (4 dấu cách mỗi cấp), sử dụng dấu {}, ngắt dòng và cách dòng hợp lý.
- Cung cấp chú thích (comment) cho chương trình, các hàm và các câu lệnh quan trọng 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 với kiểu đặt tên nhất quán.
- Tránh “hard-coding” bằng cách sử dụng khai báo/định nghĩa hằng số.

**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://github.com/nixman/mingw-builds-binaries/releases/download/14.2.0-rt\\_v12-rev0/x86\\_64-14.2.0-release-posix-seh-ucrt-rt\\_v12-rev0.7z](https://github.com/nixman/mingw-builds-binaries/releases/download/14.2.0-rt_v12-rev0/x86_64-14.2.0-release-posix-seh-ucrt-rt_v12-rev0.7z))

Sinh viên cũng cần mô tả rõ chương trình viết trong hệ điều hành nào (Windows, Linux, MacOS) trong báo cáo.

Sinh viên cố gắng tự thực hiện bài tập bằng các công cụ ở trên tránh dựa dẫm vào các công cụ AI như ChatGPT hoặc Copilot.

#### VI. Báo cáo và hướng dẫn nộp:

- Sinh viên làm bài tập 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). Khuyến khích viết bằng tiếng Anh.
- 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: giới thiệu các cấu trúc dữ liệu chính được sử dụng và/hoặc định nghĩa bởi sinh viên, 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. Nếu có quá nhiều hàm được định nghĩa trong chương trình, sinh viên có thể lựa chọn một số hàm quan trọng để mô tả.
- 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. Bảng đánh giá % đóng góp của mỗi thành viên trong bài tập lớp nếu làm theo nhóm. Ví dụ

Student names Họ và tên sinh viên	Tasks	Percentage of contribution Phần trăm đóng góp
Nguyen Van A	1, 2.1, 2.2	50%
Nguyen Van B	2.3, 2.4, 3	50%

Nếu nhóm có 2 người mà chỉ có 1 sinh viên thực hiện, thành viên còn lại không có đóng góp gì thì có thể phần trăm đóng góp là 100% và 0% tương ứng.

- Tài liệu tham khảo (nếu có).
- Sinh viên nén toàn bộ thư mục “groupId\_mini\_project\_20241” thành file “groupId\_mini\_project\_20241.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 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.
  - groupId trong tên thư mục và tên file nén thay bằng mã (ID) nhóm. Ví dụ: “20221234\_20222345\_mini\_project\_20241.zip”
- Bài làm phải được nộp qua Team Assignment đúng hạn, không nộp bài qua email hay bất cứ kênh nào khác. Mỗi nhóm chỉ cần 1 người nộp đại diện.
- Nếu có bất kỳ thắc mắc nào về đề bài sinh viên có thể liên hệ với giảng viên để làm rõ. Khuyến khích đặt câu hỏi trong Teams của lớp hơn là hỏi riêng thầy để mọi người trong lớp đều nắm được.

## VII. Đánh giá:

- Bài tập lớn sẽ được chấm như sau:
  - Hoàn thành tất cả các tasks, chương trình chạy không có lỗi gì, thực thi nhanh, xử lý tốt các trường hợp lỗi và có sáng tạo (60%).
  - Thiết kế tốt và có tính sử dụng lại cao (20%).
  - Phong cách lập trình tốt (good coding style) (20%).
  - Báo cáo trình bày đúng template, bố cục rõ ràng, trình bày dễ hiểu, không có lỗi chính tả ngữ pháp (20%).
  - Sinh viên đặt tên hoặc có cấu trúc thư mục sai quy định trong Mục III (-5%)
- Sinh viên phải tự thực hiện bài tập lớn. Không copy bài của bạn khác hoặc từ bất kỳ nguồn nào. Sinh viên phải giữ bí mật bài làm của mình. **Nếu hai hay nhiều nhóm 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 phạm quy và coi như không nộp bài không cần biết ai copy bài của ai.** Code và báo cáo copy từ nguồn khác hoặc từ công cụ AI cũng bị coi là đạo văn.
- Sinh viên chú ý nộp đúng hạn trên Team Assignment. Không được phép nộp muộn.
- Sinh viên (nhóm sinh viên) thực hiện tốt bài tập lớn sẽ được xem xét cộng điểm quá trình. Sinh viên có thể được cộng tối đa điểm thưởng chỉ khi thực hiện được cả 3 tasks và chương trình qua được 1 vài test cases.
- Các sinh viên có thể được hỏi riêng ngẫu nhiên để kiểm tra có đúng là tự làm.



- Sinh viên không nộp bài sẽ bị trừ 3 điểm quá trình. Nếu phần đóng góp của sinh viên trong nhóm quá ít cũng bị trừ điểm.

----- Hết -----