**BÁO CÁO BÀI TẬP LỚN**

**Họ và Tên:** Đồng Văn Dũng

**Mã số SV:** 20202342

Trường Điện – Điện tử

Đại học Bách khoa Hà Nội

Tp. Hà Nội, Việt Nam

**Email:** dung.dv202342@sis.hust.edu.vn

**Họ và Tên:** Trần Đỉnh Thiên

**Mã số SV:** 20202530

Trường Điện – Điện tử

Đại học Bách khoa Hà Nội

Tp. Hà Nội, Việt Nam

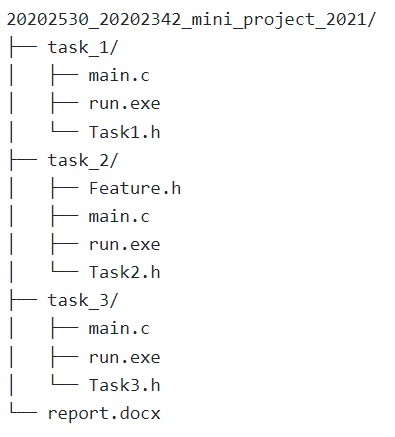
**Email:** thien.td202530@sis.hust.edu.vn

**NỘI DUNG BÁO CÁO**

1. **Ý tưởng chính.**

Các task được viết trong hệ điều hành Windows

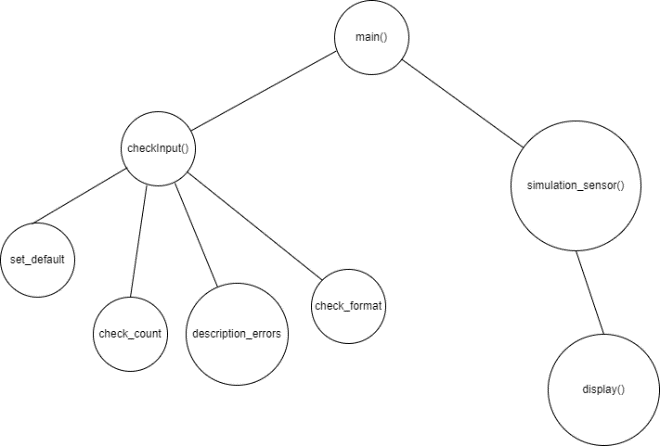
Cấu trúc chương trình:



Task 1:

* Đếm số lượng đối số đầu vào để xét các trường hợp của dữ liệu đầu vào:
* Nếu như thiếu cả 2 đối số 🡺 thiết lập giá trị mặc định
* Nếu như thiếu một trong 2 🡺 thông báo lỗi
* Nếu như đã đủ cả 2 đối số 🡺 kiểm tra định dạng dữ liệu đầu vào xem đã hợp lệ hay chưa.
* Khi mọi điều kiện đã thoả mãn 🡺 lấy thời gian hiện tại trừ đi khoảng thời gian mô phỏng để lấy được thời điểm bắt đầu mô phỏng
* Duyệt từng thời điểm mô phỏng với step là tần số trích mẫu, ghi thông tin theo định dạng vào file đầu ra.

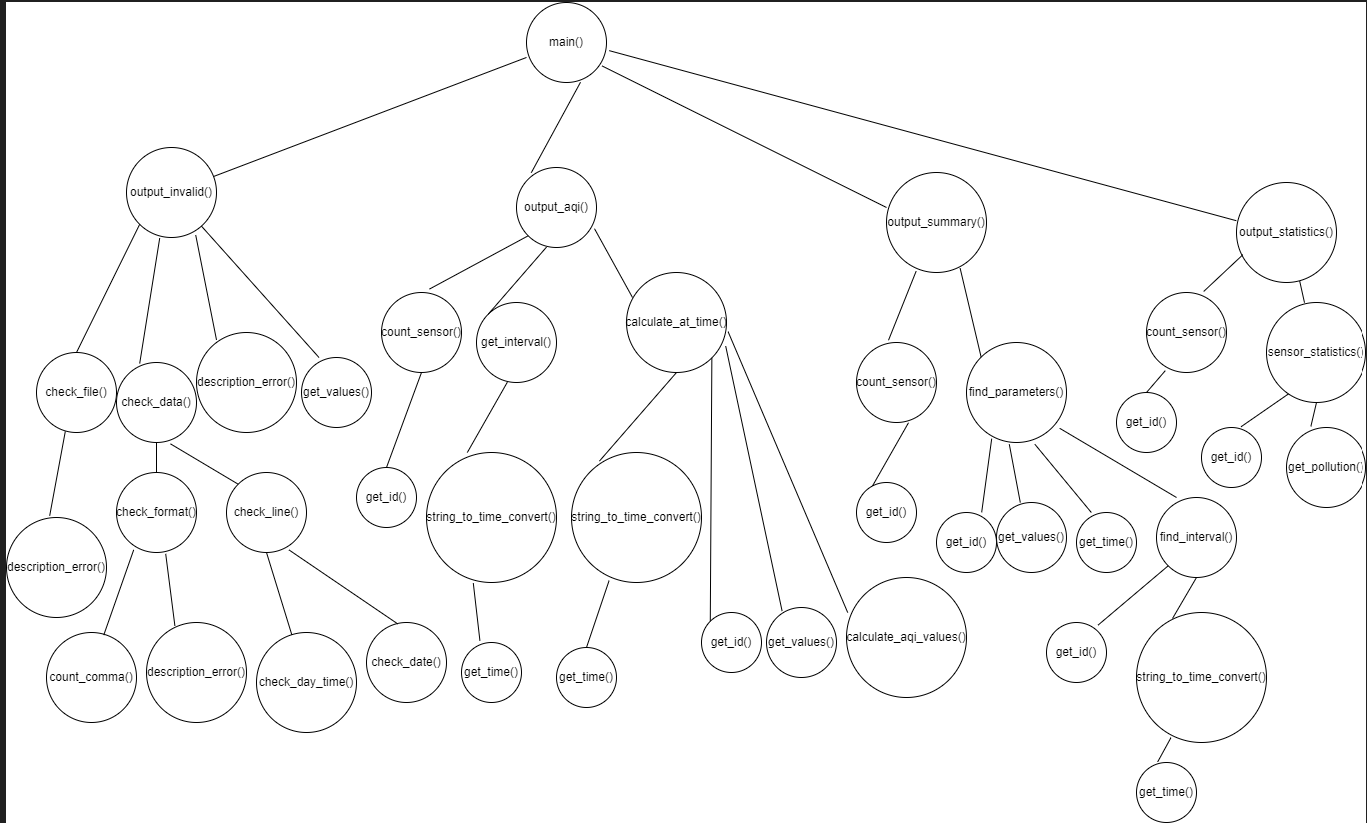
Top-down approach



Sử dụng các thư viện: stdio.h,stdlib.h,string.h,time.h,stdbool.h

Task 2:

* Kiểm tra các trường hợp lỗi của file đầu vào(định dạng file, định dạng dữ liệu,…)
* Task 2.1: Duyệt file đầu vào sau khi đã kiểm tra các lỗi,đếm các dòng dữ liệu có values không hợp lệ và ghi các dòng có values hợp lệ vào 1 file dust\_valid.csv để xử lý dữ liệu, sau đó ghi ra tổng số dòng không hợp lệ và ghi các dòng không hợp lệ vào file dust\_outlier.csv
* Task 2.2: Đếm số lượng cảm biến mô phỏng và xác định thời điểm bắt đầu và cuối cùng cần tính toán🡺 duyệt file dust\_valid.csv để tính giá trị nồng độ bụi trung bình cho từng thời điểm(từ thời điểm đầu đến cuối) 🡺 xây dựng công thức tính aqi tuyến tính theo c 🡺 lập phương trình đường thẳng aqi = a.c + b 🡺giải ra các tham số a và b ta sẽ có được công thức 🡺 ghi lại kết quả tính toán của mỗi thời điểm vào file đầu ra
* Task 2.3: Đếm số lượng cảm biến mô phỏng ,duyệt file dust\_valid.csv để tính toán cho từng cảm biến , tìm giá trị nồng độ bụi trung bình max và min của từng cảm biến , và tìm thời điểm đầu tiên tương ứng với các giá trị đó, đồng thời đếm số lần mô phỏng của cảm biến để tính giá trị trung bình đo được của mỗi cảm biến.
* Task 2.4: Đếm số lượng cảm biến mô phỏng. dùng 1 mảng để lưu trữ số lần xuất hiện của các pollution 🡺 số lần đó chính là tổng số giờ ở mỗi cấp độ ô nhiễm của từng pollution.
* Top-down approach

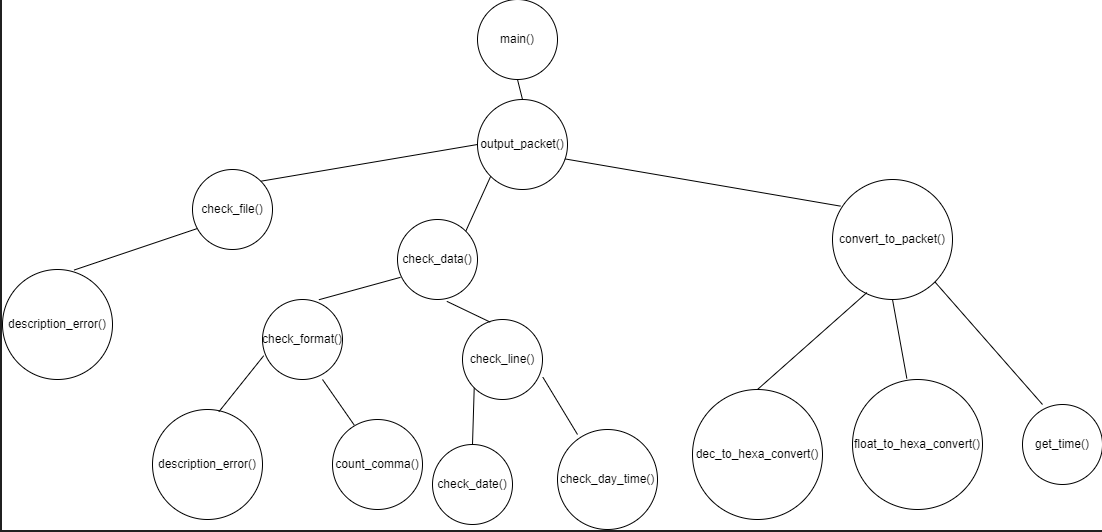


Sử dụng các thư viện: stdio.h,stdlib.h,string.h,time.h,stdbool.h

Task 3:

* Chuyển các dòng thành các gói tin có định dạng yêu cầu , thiết kế các hàm để chuyển đổi giá trị thập phân thành hexa và số thực thành hexa theo chuẩn IEEE 754 🡺 ghi các gói tin đó vào file đầu ra.

Top-down approach



Sử dụng các thư viện: stdio.h,stdlib.h,string.h,time.h,stdbool.h

1. **Thiết kế chi tiết.**

Task 1:

* Hàm: int main(int argc, char \*argv[]);

Tên hàm: main

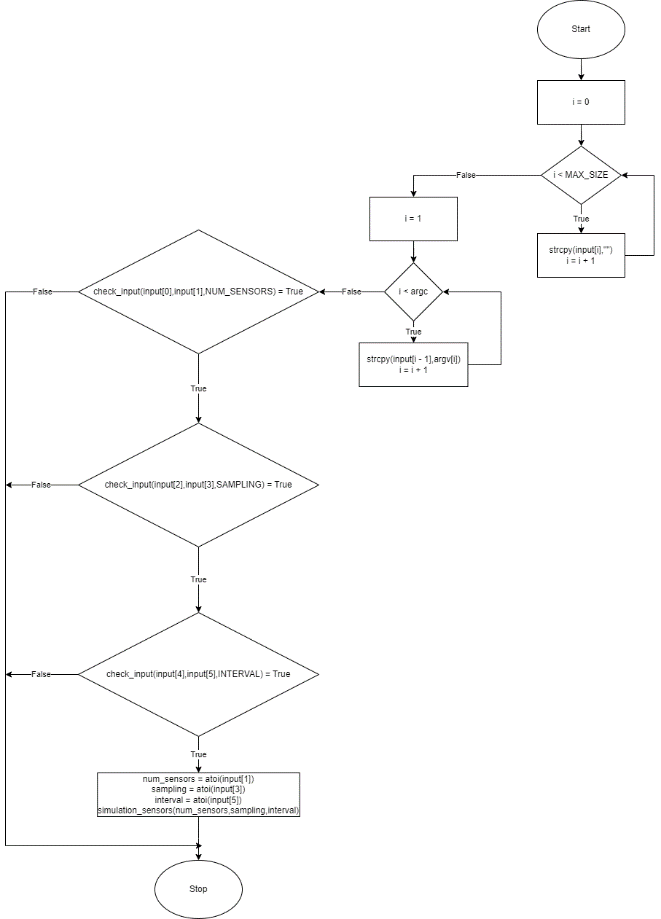
Kiểu trả về: int

Danh sách tham biến: argc, argv

Đầu vào: argv (chuỗi chứa các đối số đầu vào, gồm các flag và giá trị tương ứng với các flag)

Đầu ra: Dữ liệu được ghi vào file đầu ra dust\_sensor.csv

Flowchart:



* Hàm: void simulation\_sensors(int num\_sensors,int sampling,int interval)

Tên hàm: simulation\_sensors

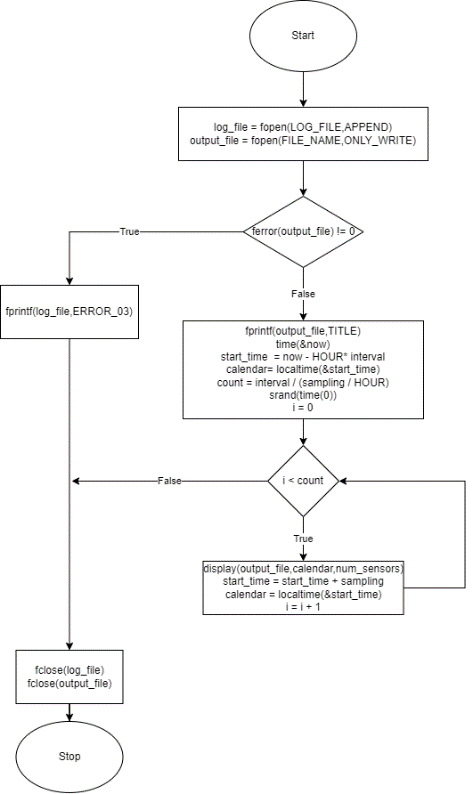
Kiểu trả về: void

Danh sách tham biến: num\_sensors,sampling,interval

Đầu vào: num\_sensors(số lượng cảm biến),sampling(tần số trích mẫu),interval(khoảng thời gian mô phỏng)

Đầu ra: Dữ liệu được ghi vào file đầu ra dust\_sensor.csv

Flowchart:



Task 2:

* Hàm: int main(int argc, char \*argv[]);

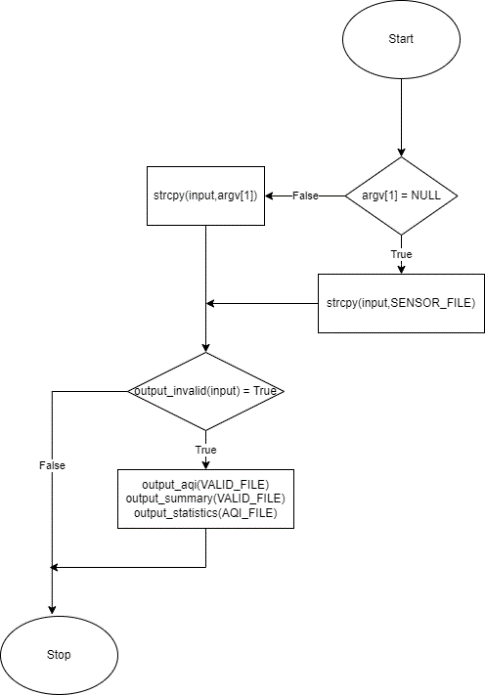
Tên hàm: main

Kiểu trả về: int

Danh sách tham biến: argc, argv

Đầu vào: argv (chuỗi chứa các đối số đầu vào, gồm liên kết đến file csv cần xử lý)

Đầu ra: Dữ liệu được ghi vào các file đầu ra tương ứng.

Flowchart: 

* Hàm: void output\_statistics(char \*link);

Tên hàm: output\_statistics

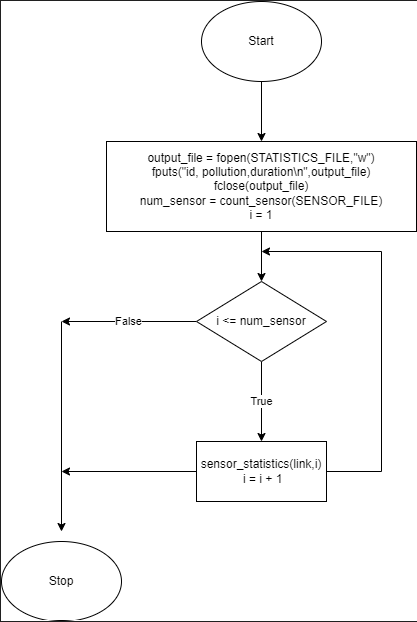
Kiểu trả về: void

Danh sách tham biến: link

Đầu vào: link(liên kết đến file dữ liệu đầu vào)

Đầu ra: Dữ liệu được ghi vào file đầu ra dust\_statistics.csv

Flowchart:



Task 3:

* Hàm: int main(int argc, char \*argv[]);

Tên hàm: main

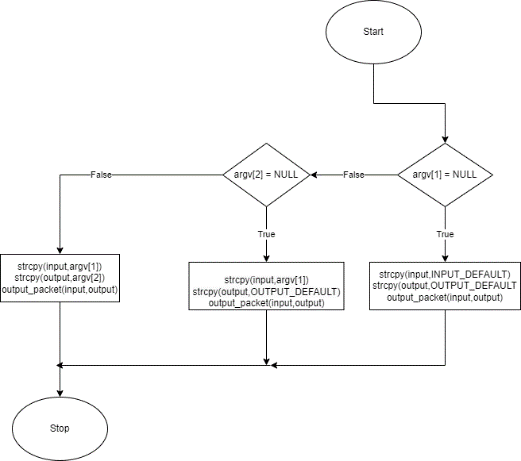
Kiểu trả về: int

Danh sách tham biến: argc, argv

Đầu vào: argv (chuỗi chứa các đối số đầu vào, gồm liên kết đến file csv đầu vào)

Đầu ra: Dữ liệu được ghi vào file đầu ra tương ứng.

Flowchart:



* Hàm: void output\_packet(char\* input,char\* output);

Tên hàm: output\_packet

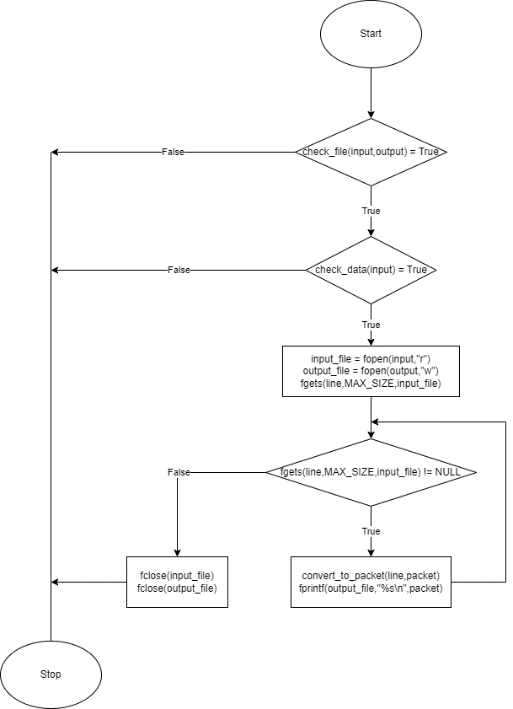
Kiểu trả về: void

Danh sách tham biến: input,output

Đầu vào: input(liên kết tời file đầu vào),output(liên kết tới file đầu ra)

Đầu ra: Dữ liệu được ghi vào file dầu ra tương ứng.

Flowchart:



1. **Kết quả và đánh giá.**
   1. Kết quả chạy chương trình.
   * Chương trình chạy không có lỗi gì
   * Dữ liệu hoàn toàn đúng như yêu cầu đề bài
   * Tất cả các trường hợp lỗi đều được thông báo nếu xảy ra
   1. Đánh giá chất lượng chương trình.

* Chạy đúng và không có bug
* Một số thuật toán chưa được thực sự tối ưu
* Thời gian chạy chương trình khá nhanh.

1. **Kết luận**

* Đã làm được:
* Hoàn thành tất cả cá task
* Xử lý tất cả các lỗi đã được đề xuất
* Mã gọn gang, ngắn gọn ,dễ theo dõi
* Cung cấp chú thích rõ ràng,dễ hiểu
* Tên hàm ,tên biến sử dụng hoàn toàn bằng tiếng Anh
* Tránh được “hard-coding”
* Coding style nhất quán theo quy định GNU
* Một số thuật toán đã được tối ưu
* Chưa làm được:
* Một số thuật toán trong chương trình chưa thực sự được tối ưu
* Còn một số chỗ chưa được tối ưu.
* Đánh giá % đóng góp của các thành viên:
* Trần Đỉnh Thiên: 50%
* Đồng Văn Dũng: 50%

1. **Tài liệu tham khảo**
   1. Slide kỹ thuật lập trình chương 1,2,3
   2. Website:https://www.geeksforgeeks.org/c-programming-language/?ref=ghm
   3. Website:https://www.programiz.com/c-programming
   4. Website:https://www.tutorialspoint.com/cprogramming/index.htm