**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN LẬP TRÌNH TÍNH TOÁN**

**TÊN ĐỀ TÀI 705:**

**MÔ PHỎNG BÀI TOÁN TÔ MÀU BẢN ĐỒ BẰNG TÔ MÀU CÁC ĐỈNH CỦA ĐỒ THỊ**

Người hướng dẫn**:  TRẦN HỒ THỦY TIÊN**

Sinh viên thực hiện**:**

**Trần Nhật Minh** **LỚP: 20CLC\_Nhat1   NHÓM: 20NH10**

**Phạm Tiến Hữu** **LỚP: 20CLC\_Nhat1   NHÓM: 20NH10**

**Đà Nẵng, 04/2020**

MỤC LỤC

[MỤC LỤC 1](#_Toc80733084)

[DANH MỤC HÌNH VẼ 3](#_Toc80733085)

[LỜI MỞ ĐẦU 5](#_Toc80733086)

[1. Mục đích thực hiện đề tài 6](#_Toc80733087)

[2. Mục tiêu đề tài 6](#_Toc80733088)

[3. Phạm vi và đối tượng nghiên cứu 6](#_Toc80733089)

[4. Phương pháp nghiên cứu 6](#_Toc80733090)

[5. Cấu trúc của đồ án môn học 6](#_Toc80733091)

[1. TỔNG QUAN ĐỀ TÀI 8](#_Toc80733092)

[2. CƠ SỞ LÝ THUYẾT 8](#_Toc80733093)

[2.1. Ý tưởng 8](#_Toc80733094)

[2.2. Cơ sở lý thuyết 9](#_Toc80733095)

[3. TỔ CHỨC CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN 10](#_Toc80733096)

[3.1. Phát biểu bài toán 10](#_Toc80733097)

[**3.1.** 10](#_Toc80733101)

[3.2. Cấu trúc dữ liệu 10](#_Toc80733102)

[3.2.1. Arrays 10](#_Toc80733103)

[3.2.2. Graph 11](#_Toc80733104)

[3.2.3. Thao tác với file 13](#_Toc80733105)

[3.2.4. Thuật toán sắp xếp selection sort 16](#_Toc80733106)

[3.2.5. Đồ hoạ 17](#_Toc80733109)

[3.2.5.1. Lý thuyết thư viện đồ hoạ 17](#_Toc80733110)

[3.2.5.2. Cách tính toạ độ trong đồ thị 22](#_Toc80733111)

[3.3. Thuật toán 24](#_Toc80733112)

[4. CHƯƠNG TRÌNH VÀ KẾT QUẢ 38](#_Toc80733113)

[4.1. Tổ chức chương trình 38](#_Toc80733114)

[4.2. Ngôn ngữ cài đặt 38](#_Toc80733115)

[4.2.1. Giới thiệu 38](#_Toc80733116)

[4.2.2. Tổng quan 39](#_Toc80733117)

[4.3. Kết quả 40](#_Toc80733118)

[4.3.1. Giao diện chính của chương trình 40](#_Toc80733119)

[4.3.2. Kết quả thực thi của chương trình 42](#_Toc80733120)

[4.4. Nhận xét đánh giá 44](#_Toc80733121)

[5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 45](#_Toc80733122)

[5.1. Kết luận 45](#_Toc80733123)

[5.1. Hướng phát triển 45](#_Toc80733124)

[TÀI LIỆU THAM KHẢO 46](#_Toc80733125)

[PHỤ LỤC 47](#_Toc80733126)

DANH MỤC HÌNH VẼ

Hình 1: Cấu trúc mảng 11

Hình 2: Đồ thị có hướng. 12

Hình 3: Đồ thị vô hướng. 12

Hình 4: Minh hoạ thuật toán selection sort. 16

Hình 5: Màn hình đồ hoạ 18

Hình 6: Bảng 16 màu trong thư viện đồ hoạ C/C++ 18

Hình 7: Bảng các hàm thường dùng thư viện <graphics.h> (part 1) 19

Hình 8: Bảng các hàm thường dùng thư viện <graphics.h> (part 1) 20

Hình 9: Console bị lỗi do các đoạn thẳng dính liền với nhau 22

Hình 10: Console tô màu hoàn chỉnh (đã loại bỏ đoạn dư thừa của các đoạn thẳng). 23

Hình 11: Sơ đồ cấu trúc chương trình 24

Hình 12: Sơ đồ khối main 25

Hình 13: Sơ đồ khối hàm Fileread 26

Hình 14: Sơ đồ khối hàm Introduce 27

Hình 15: Sơ đồ khôi hàm Degree 28

Hình 16: Sơ đồ khối hàm Sort 29

Hình 17: Sơ đồ khối hàm Paint 30

Hình 18: Sơ đồ khối hàm Check 31

Hình 19: Sơ đồ khối hàm Delete 32

Hình 20: Sơ đồ khối hàm Out 33

Hình 21: Sơ đồ khối hàm Graphics 34

Hình 22: Sơ đồ khối hàm draw\_line 35

Hình 23: Sơ đồ khối hàm draw\_vertex 36

Hình 24: Giao diện của chương trình lúc ban đầu 39

Hình 25: Không tìm thấy file “Introduce.txt” 39

Hình 26: Giao diện của chương trình sau khi nhấn Enter 40

Hình 27: Không tìm thấy file “dothi.txt” 40

Hình 28: Dừng file khi thấy đồ thị chưa đúng 41

Hình 29: Kết quả thu được (toàn màn hình) 41

Hình 30: Console 1 (kết quả tô màu) 42

Hình 31: Console 2 (phác hoạ đồ thị) 42

LỜI MỞ ĐẦU

**---0O0---**

Đồ thị phẳng là một trong những đồ thị đặc biệt nhận được nhiều sự quan tâm trong lý thuyết đồ thị do cấu trúc đặc biệt và ứng dụng rộng khắp của nó. Tô màu bản đồ là một trong những bài toán cơ bản của Lý thuyết đồ thị phẳng, được ứng dụng nhiều trong tin học, nhằm giải quyết các vấn đề thực tiễn liên quan đến phân hoạch, phân nhóm, bản đồ.

Qua chương trình “Tô màu các đỉnh của đồ thị”, hy vọng sẽ giúp các bạn có một khái niệm cơ bản về đồ thị phẳng, tạo cho các bạn sự hứng thú trong việc tìm hiểu thêm về Lý thuyết đồ thị. Đồng thời nhóm chúng mình mong rằng sẽ có thể dùng chương trình này để phát triển nhiều thuật toán có nhiều ứng dụng hơn trong tương lai.

Nhóm chúng mình xin gửi lời cảm ơn sâu sắc đến ThS.Trần Hồ Thủy Tiên, người đã hướng dẫn chúng em rất nhiều trong suốt quá trình tìm hiểu và hoàn thành đồ án này. Đồng thời chúng mình cũng xin chân thành cảm ơn các thầy cô trong bộ môn cũng như các thầy cô trong trường đã trang bị cho em những kiến thức nền tảng vô cùng quan trọng.

Trong quá trình nghiên cứu và phát triển có thể sẽ có những sai sót, khiếm khuyết. Hy vọng nhóm chúng mình sẽ nhận được sự đóng góp ý kiến của các bạn đọc và các quý thầy cô để nhóm chúng mình khắc phục và để chương trình ngày càng hoàn thiện hơn.

**Xin chân thành cảm ơn!**

# Mục đích thực hiện đề tài

* Nghiên cứu và vận dụng các học phần, các kiến thức đã học cần thiết để xây dựng, phát triển một ứng dụng giải quyết và đáp ứng các bài toán, vấn đề trong thực tế.
* Củng cố, nâng cao được kỹ năng phân tích xây dựng thuật toán, giải quyết vấn đề và kỹ năng lập trình.
* Phát triển một số kỹ năng mềm (làm việc nhóm, tra cứu, phân tích).

# Mục tiêu đề tài

* Viết được chương trình giải quyết bài toán đặt ra bằng ngôn ngữ lập trình  
  C/C++.
* Giúp sinh viên phát triển và rèn luyện các kỹ năng phân tích yêu cầu bài toán, kỹ năng giải quyết vấn đề, kỹ năng tư duy lập trình. Đồng thời rèn luyện và bổi dưỡng thêm những kỹ năng mềm đối với người kỹ sư ngành CNTT.
* Sử dụng được các mảng đa chiều để lưu dữ liệu về ma trận các liên kết giữa các đỉnh, các mảng một chiều để lưu bậc của các đỉnh, màu sắc ứng với các đỉnh; các thuật toán thao tác với mảng, các thao tác với file.

1. Phạm vi và đối tượng nghiên cứu

* Nghiên cứu và tìm hiểu các phương pháp, cấu trúc dữ liệu, các kiến thức liên quan đến sắp xếp mảng, cách đọc dữ liệu từ 1 file, các thư viện đồ hoạ… để áp dụng vào chương trình thực hiện các yêu cầu của đề tài.
* Sử dụng các kiến thức từ các học phần áp dụng, xây dựng thuật toán. Tìm hiểu các vấn đề, các yêu cầu cần thiết của một thuật toán tô màu bản đồ để xây dựng một chương trình giải quyết các vấn đề đó

1. Phương pháp nghiên cứu

* Dựa trên các kiến thức cơ bản đã học trên lớp và tham khảo, nghiên cứu thêm tài liệu về lập trình bằng ngôn ngữ C/C++.
* Tìm hiểu các yêu cầu, chức trong thực tế của ứng dụng này, tiến hành phân tích, tính toán, phác thảo, tiến hành viết chương trình theo yêu cầu được đặt ra.

1. Cấu trúc của đồ án môn học

* Trang bìa.
* Trang bìa trong.
* Trang Mục lục.
* Danh mục hình vẽ.
* Trang “Lời mở đầu”.
* Tổng quan đề tài.
* Cơ sở lý thuyết.
* Ý tưởng.
* Cơ sở lý thuyết.
* Tổ chức cấu trúc dữ liệu và thuật toán
* Phát biểu bài toán. Cấu trúc dữ liệu.
* Thuật toán.
* Chương trình và kết quả
* Tổ chức chương trình.
* Ngôn ngữ cài đặt.
* Kết quả.
* Kết luận và hướng phát triển.
* Trang Tài liệu tham khảo.
* Các trang Phụ lục.

1. TỔNG QUAN ĐỀ TÀI

Trên các bản đồ, các miền khác nhau (miền ở đây được hiểu là các quốc gia trên bản đồ thế giới hay các tỉnh trong một bản đồ hành chính quốc gia) được tô màu sao cho 2 miền có chung biên giới không trùng màu nhau. Đối với bản đồ có nhiều miền, nếu ta dùng một số lượng lớn màu thì sẽ rất khó phân biệt các miền có màu gần giống nhau, vì thế người ta chỉ dùng một số lượng màu nhất định để tô màu bản đồ. Một bài toán được đặt ra là: có thể dùng ít nhất bao nhiêu màu để tô màu một bản đồ sao cho các miền kề nhau không cùng một màu.

Bài toán này dẫn đến định lý bốn màu nổi tiếng và định lý năm màu. Các dạng bài toán tô màu bản đồ có thể áp dụng *Thuật toán tô màu Greedy* để tìm ra số màu ít nhất để tô cho các miền trên bản đồ.

Chương trình “Tô màu các đỉnh của đồ thị” được sinh ra nhằm mục đích mô phỏng lại đôi chút của *thuật toán tô màu Greedy*. Nhằm giúp bạn đọc có cái nhìn khách quan về lý thuyểt đồ thị cũng như là hiểu thêm về cách chứng minh định lý bốn màu – định lý lớn đầu tiên được chứng minh băng máy vi tính.

* Mô tả chung:
  + Nhập dữ liệu vào file “dothi.txt”: Dữ liệu phải chứa số đỉnh của đồ thị và ma trận các liên kết của đồ thị.
  + Chương trình sẽ nạp dữ liệu từ fỉle trên vào các mảng.
  + Bằng những thuật toán sắp xếp thì sẽ cho ra kết quả các màu được tô lên các đỉnh của đồ thị sao cho số màu là ít nhất

Sau khi chạy chương trình thành công: sẽ hiện ra 1 console cho ra đồ thị đã nhập vào và kết quả các màu được tô tương ứng với các đỉnh, và 1 console hiển thị hình vẽ phác thảo của đồ thị đó.

1. CƠ SỞ LÝ THUYẾT
   1. Ý tưởng

Áp dụng các kiến thức về mảng, đặc biệt là sắp xếp mảng, thao tác với file kết hợp với lý thuyết về đồ thị để xây dựng thuật toán và các hàm về đồ hoạ viết chương trình, thực hiện các yêu cầu đề ra. Các nội dung lần lượt gồm:

* Đọc dữ liệu đầu vào từ file “Introduce.txt”, “dothi.txt” bằng các hàm thao tác với file.
* Dùng dữ liệu đầu vào để khởi tạo các mảng.
* Sắp xếp các mảng lại theo bậc của các đỉnh.
* In ra kết quả của các đỉnh tương ứng với màu được tô.
* Dùng các kết quả thu được phác hoạ thành 1 đồ thị bằng các hàm đồ hoạ.

à Từ đó ta dần dần xây dựng hoàn thiện thuật toán và các chức năng của ứng dụng bằng ngôn ngữ lập trình C.

* 1. Cơ sở lý thuyết

Nghiên cứu từ các tài liệu, các nguồn tham khảo, từ các học phần đã được học, chúng em sử dụng các kiến thức, các chức năng được sử dụng trong C, từ đó phân tích, xây dựng thuật toán, một chương trình hoàn chỉnh:

* Sử dụng các thư viện để bổ trợ cho các chức năng muốn được sử dụng trong chương trình:
* Thư viện <stdio.h>: Cung cấp cốt lõi của những khả năng nhập trong C. Tập tin này bao gồm họ hàm printf.
* Thư viện <graphics.h>: Dùng cho việc phác hoạ đồ thị bao gồm các hàm initwindow, line, circle, setfillstyle, floodfill:
* initwindow (int width, int height): Khởi tạo console đồ hoạ với chiều cao chiều rộng được truyền vào.
* Line (int x1, int y1, int x2, int y2): Vẽ 1 đường thẳng bắt đầu từ điểm có toạ độ (x1, y1) tới điểm có toạ độ (x2, y2) trên console được khởi tạo.
* circle (int x, int y, int radius): Vẽ 1 hình tròn có tâm tại điểm có toạ độ (x, y) và có bán kính là radius.
* setfillstyle (int pattern, int color): chọn màu để tô vào các đỉnh ứng với pattern là kiểu tô và color là loại màu.
* floodfill (int x, int y, int border): Chọn khu vực kín mà mình muốn tô màu, (x, y) là toạ độ điểm bên trong khu vực cần tô và border là màu sắc của phần viền của vùng mà cần tô màu (tương tự như cách tô màu bằng phần mềm paint).
* settextstyle (int font, int direction, int charsize): Vì font mặc định rất bé nên chỉnh qua một font khác sẽ khiến cho việc hiển thị rõ ràng hơn. Font sử dụng trong chương trình là SANS\_SERIF\_FONT.
* Thư viện <conio.h>: Để không cho chương trình tắt làm tắt luôn đi console đồ hoạ bằng hàm getch ().
* Thư viện <math.h>: Để tính toán toạ độ của các điểm trên console đồ hoạ với các hàm cần dùng đến như là atan (), cos(), sin().
* Vận dụng các chức năng liên quan đến các phần như là: thao tác với file, mảng…
* Thao tác với file:

Mở file: dùng hàm fopen ().

Lấy dữ liệu từ file: dùng hàm fgetc (), fscanf ().

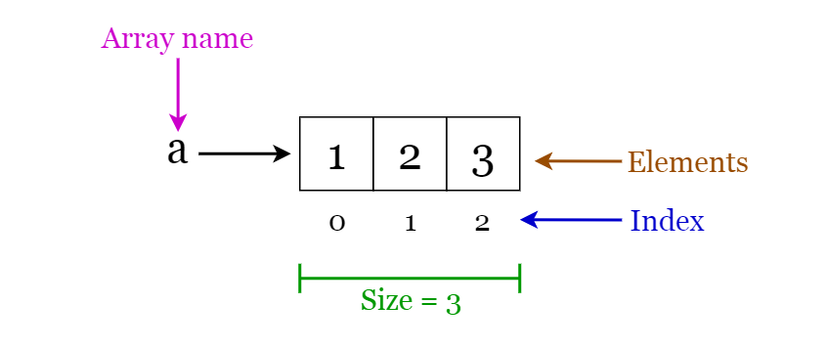
Đóng file: dùng hàm fclose ().

1. TỔ CHỨC CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN
   1. Phát biểu bài toán

* Đối với bất kỳ mặt phẳng nào được chia thành các vùng phân biệt, chẳng hạn như bản đồ hành chính của một quốc gia, chỉ cần dùng tối đa bốn màu để phân biệt các vùng lân cận với nhau. Hai vùng được coi là lân cận nếu như chúng có chung nhau một đoạn đường biên, không tính chung nhau một điểm.
* Để giải bài toán: Xây dựng các đồ thị tương ứng với các bản đồ và đưa giả thuyết 4 màu về giả thuyết số màu của đồ thị phẳng.
* Với mỗi bản đồ, ta có thể định nghĩa một đồ thị:
* Tập đỉnh: gồm các đỉnh mà mỗi đỉnh tương ứng với một vùng trên bản đồ;
* Tập cạnh: mỗi cạnh tương ứng với 2 vùng kề nhau có chung một biên giới.
* Cách tô màu vùng của bản đồ tương ứng với cách tô màu đỉnh của đồ thị, sao cho hai đỉnh kề nhau không cùng một màu được sử dụng ít nhất.
* Input: đồ thị G = (V, E).
* Output: đồ thị G = (V, E) có các đỉnh đã được tô màu.

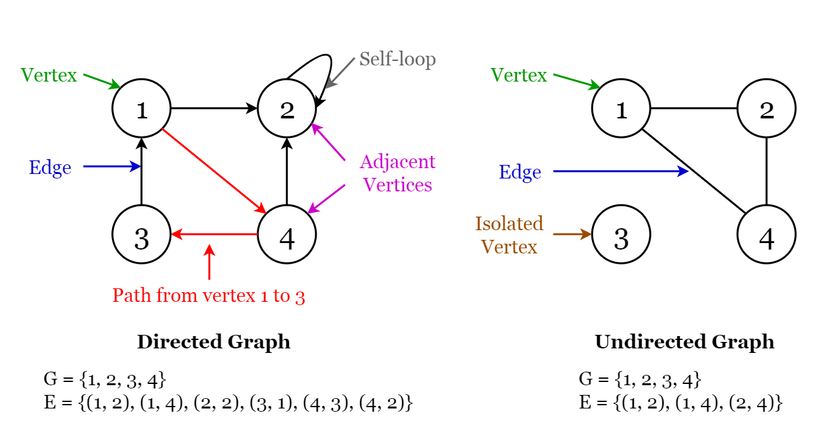
3. 2. Cấu trúc dữ liệu
      1. Arrays

* Một Array - mảng là một cấu trúc với kích thước cố định, có thể giữ các item có cùng kiểu dữ liệu. Nó có thể là một mảng các số nguyên, một mảng các số thực, một mảng string hay kể cả một mảng của các mảng (mảng 2 chiều). Mảng được đánh chỉ mục, cho phép ta có thể truy cập ngẫu nhiên vào mảng.

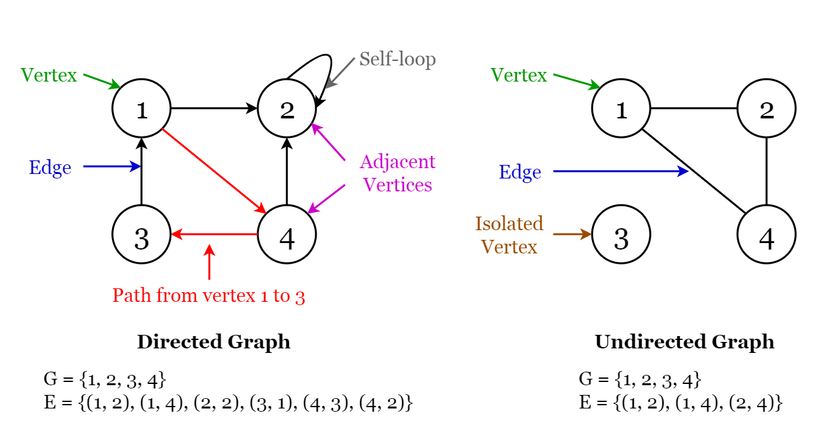
***Hình 1. Cấu trúc mảng***

* Các phép toán trên mảng:
* Truy cập ngẫu nhiên: Cho phép ta truy cập tới một phần tử bất kỳ trong mảng bằng chỉ mục (index) của nó.
* Tìm kiếm: Tìm kiếm một phần tử trong mảng bằng chỉ mục hoặc giá trị của nó.
* Cập nhật: Cập nhật giá trị của một phần tử đã tồn tại trong mảng thông qua chỉ mục của nó. Việc chèn thêm phần tử hay xoá bớt phần tử trên mảng là không thể thực hiện thẳng bởi mảng có kích thước cố định. Nếu muốn thêm một phần tử vào mảng, đầu tiên ta phải tạo một mảng mới với kích thước được tăng lên so với mảng cũ, sau đó copy các phần từ và thêm phần tử mới vào mảng mới. Tương tự với việc xoá phần tự, ta cần một mảng mới với kích thước nhỏ hơn.
* Ứng dụng của mảng:
* Sử dụng để xây dựng nên các cấu trúc dữ liệu khác như là array list, heap, hash table, vector hay matrix.
* Sử dụng cho các thuật toán sắp xếp khác nhau như là insertion sort, quick sort, bubble sort, merge sort.
  + 1. Graph
* Một graph - đồ thị là một tập hợp hữu hạn các đỉnh (nút) và đường đi giữa các đỉnh này. Kích thước của một đồ thị được tính bằng số lượng, bậc của đồ thị được tính bằng số đỉnh của nó. Hai đỉnh được gọi là liền kề nếu chúng được nối với nhau bởi cùng một đường.
* Đồ thị có hướng:

Một đồ thị gọi là đồ thị có hướng nếu tất cả đường đi trên nó đều được đánh dấu chiều giữa điểm đầu và điểm cuối. Ký hiệu (u,v) là đường đi từ đỉnh u tới đỉnh v. Sefl-loop: việc một đỉnh có đường đi tới chính nó.

***Hình 2. Đồ thị có hướng*** 

* Đồ thị vô hướng:

Một đồ thị gọi là đồ thị vô hướng nếu tất cả đường đi trên nó đều không quy định chiều. Nếu một đỉnh trong đồ thị không được kết nối tới bất kỳ đỉnh nào, ta nói nó bị cô lập.

***Hình 3. Đồ thị vô hướng***

* Ứng dụng của đồ thị
* Sử dụng để biểu diễn mạng xã hội. Mỗi người dùng là một định và người dùng sẽ được kết nối với nhau bởi các cạnh.
* Sử dụng để biểu diễn các trang web và link trong các công cụ tìm kiếm. Trang web trên mạng internet sẽ được liên kết tới các trang khác qua các đường link. Mỗi trang web là một đỉnh và các link sẽ là các cạnh.
* Sử dụng để biểu diễn vị trí và đường đi trong GPS. Vị tí là các đỉnh và đường đi giữa các vị trí chính là các cạnh. Được sử dụng để tính toán đường đi ngắn nhất giữa 2 địa điểm.
  + 1. Thao tác với file
* Tại sao chúng ta cần đến file?
* Dữ liệu được lưu ở biến của chương trình, và nó sẽ biến mất khi chương trình kết thúc. Sử dụng file để lưu trữ dữ liệu cần thiết để đảm bảo dữ liệu của chúng ta không bị mất ngay cả khi chương trình của chúng ta ngừng chạy.
* Nếu chương trình có đầu vào (input) là lớn, sẽ rất vất vả nếu phải nhập mỗi khi chạy. Thay vào đó, lưu vào file sẽ giúp chương trình tự đọc mỗi lần khởi chạy.
* Dễ dàng sao chép, di chuyển dữ liệu giữa các thiết bị với nhau.
* Các kiểu file:

File văn bản – text file

File nhị phân – binary file

* Các thao tác với file:
* Khi làm việc với file, ta cần khai báo 1 con trỏ kiểu FILE. Việc khai báo này là cần thiết để có sự kết nối giữa chương trình và tập tin cần thao tác.

FILE \*fptr;

* Thao tác mở file:

Để đọc ghi file trong C cũng như trong mọi ngôn ngữ lập trình, việc đầu tiên cần làm là mở file mà ta muốn làm việc.

Trong ngôn ngữ lập trình C/C++, chúng ta có thể mở file bằng cách sử dụng hàm fopen() như sau:

fptr = fopen("fileopen","mode");

Trong đó mode là một tham số chúng ta cần chỉ định

Dưới đây là các giá trị có thể có của tham số mode  nói trên:

|  |  |
| --- | --- |
| **Mode** | **Ý nghĩa** |
| r | Mở file chỉ cho phép đọc. |
| rb | Mở file chỉ cho phép đọc dưới dạng nhị phân. |
| w | Mở file chỉ cho phép ghi. |
| wb | Mở file chỉ cho phép ghi dưới dạng nhị phân. |
| a | Mở file ở chế độ ghi “append”. Tức là sẽ ghi vào cuối của nội dung đã có. |
| ab | Mở file ở chế độ ghi nhị phân “append”. Tức là sẽ ghi vào cuối của nội dung đã có. |
| r+ | Mở file cho phép cả đọc và ghi. |
| rb+ | Mở file cho phép cả đọc và ghi ở dạng nhị phân. |
| w+ | Mở file cho phép cả đọc và ghi. |
| wb+ | Mở file cho phép cả đọc và ghi ở dạng nhị phân. |
| a+ | Mở file cho phép đọc và ghi “append”. |
| ab+ | Mở file cho phép đọc và ghi “append” ở dạng nhị phân. |

* Đọc và ghi file trong C/C++:
* Đọc và ghi vào file text

Để đọc và ghi dữ liệu vào file text, ta sử dụng hai hàm fprintf() và fscanf().

**Ví dụ 1:** Ghi dữ liệu vào file text

#include <stdio.h>

#include <stdlib.h>

int main()

{

   int num;

   FILE \*fptr;

   fptr = fopen("C:\\program.txt","w");

   if(fptr == NULL)

   {

      printf("Error!");

      exit(1);

   }

   printf("Enter num: ");

   scanf("%d",&num);

   fprintf(fptr,"%d",num);

   fclose(fptr);

   return 0;

}

Chương trình này lấy một số từ người dùng và lưu trữ trong file program.txt. Sau khi biên dịch và chạy chương trình này, ta có thể thấy file program.txt được tạo trong ổ C của máy tính. Khi mở file ta có thể thấy số nguyên đã nhập.

**Ví dụ 2:** Đọc dữ liệu từ file text

#include <stdio.h>

#include <stdlib.h>

int main()

{

   int num;

   FILE \*fptr;

   if ((fptr = fopen("C:\\program.txt","r")) == NULL){

       printf("Error! opening file");

       exit(1);

   }

   fscanf(fptr,"%d", &num);

   printf("Value of n=%d", num);

   fclose(fptr);

   return 0;

}

Chương trình này đọc số nguyên có trong file program.txt và in ra màn hình.

Nếu đã tạo thành công file từ ví dụ 1 thì khi chạy chương trình này sẽ nhận được số nguyên đã nhập.

Các hàm khác như fgetchar(), fputc(), v.v. sử dụng tương tự theo cách này.

* Đọc và ghi dữ liệu vào file binary

Hai hàm fread() và fwrite() được dùng để đọc và ghi vào file ở định dạng nhị phân binary.

*Ghi file nhị phân:*

Để ghi vào file nhị phân, cần sử dụng hàm fwrite(). Hàm này có bốn tham số:

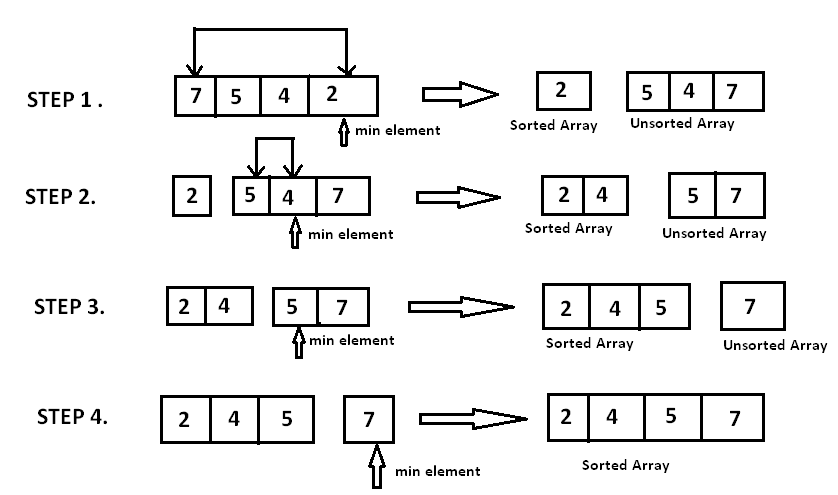
1. Đường dẫn đến file cần ghi
2. Kích thước của dữ liệu
3. Số loại dữ liệu như vậy
4. Con trỏ đến file mà bạn muốn ghi

fwrite(address, size, numbers, pointerToFile);

*Đọc file nhị phân:*

Để đọc file nhị phân thì ta sử dụng hàm fread(), hàm này cũng có bốn tham số như hàm fwrite(),

fread(address, size, numbers, pointerToFile);

* + 1. Thuật toán sắp xếp selection sort
* Thuật toán selection sort sắp xếp một mảng bằng cách đi tìm phần tử có giá trị nhỏ nhất (giả sử với sắp xếp mảng tăng dần) trong đoạn đoạn chưa được sắp xếp và đổi cho phần tử nhỏ nhất đó với phần tử ở đầu đoạn chưa được sắp xếp(không phải đầu mảng). Thuật toán sẽ chia mảng làm 2 mảng con:
* Một mảng con đã được sắp xếp
* Một mảng con chưa được sắp xếp
* Tại mỗi bước lặp của thuật toán, phần tử nhỏ nhất ở mảng con chưa được sắp xếp sẽ được di chuyển về đoạn đã sắp xếp.

***Hình 4. Minh họa thuật toán selection sort***

* Code minh họa:

void selectionSort(int arr[], int n)

{

    int i, j, min\_idx;

    for (i = 0; i < n-1; i++)

    {

    min\_idx = i;

    for (j = i+1; j < n; j++)

        if (arr[j] < arr[min\_idx])

        min\_idx = j;

         swap(arr[min\_idx], arr[i]);

    }

}

void swap(int &xp, int &yp)

{

    int temp = xp;

    xp = yp;

    yp = temp;

}

* 2. 1. Đồ hoạ

Do thư viện đồ họa graphics và winbgim không được tích hợp sẵn trong thư viện của phần mềm Dev C/ C++ nên ta phải tự thêm 2 thư viện đồ họa nếu muốn sử dụng để lập trình đồ họa.

* + - 1. Lý thuyết thư viện đồ hoạ

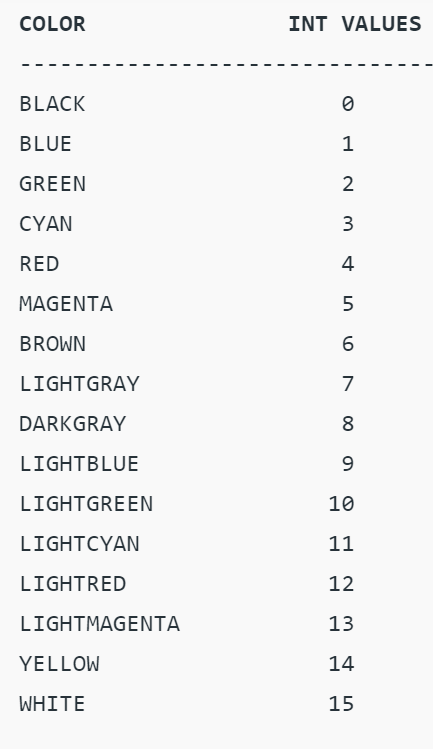
**graphics.h** hay tên chính xác và đầy đủ là Borland Graphics Interface (BGI) là một thư viện đồ họa rất phổ biến trên DOS và các máy tính chạy hệ điều hành Windows thời kì đầu như Windows 95, Windows 98.

Thư viện này cung cấp cho người dùng 2 file: graphics.h và graphics.lib để có thể sử dụng được với C/C++ cũng như module graph nếu người dùng sử dụng ngôn ngữ Pascal. Bộ thư viện này đi kèm với IDE Borland C++ 3.1 (1992).

Một trong những điểm mạnh của thư viện này là việc khởi tạo cũng như sử dụng rất đơn giản, vì vậy dù ra đời rất lâu nhưng hiện tại vẫn có rất nhiều trường đại học sử dụng cho mục đích giảng dạy.



***Hình 5. Màn hình đồ hoạ.***



***Hình 6. Bảng 16 màu trong thư viện đồ hoạ của C/C++.***

- Khởi động chế độ đồ hoạ:

initwindow (int x, int y): Kích thước màn hình đồ hoạ

VD: initwindow (900,700);

- Kiểm tra xem có lỗi gì khi tạo màn hình đồ hoạ:

graphresult ();

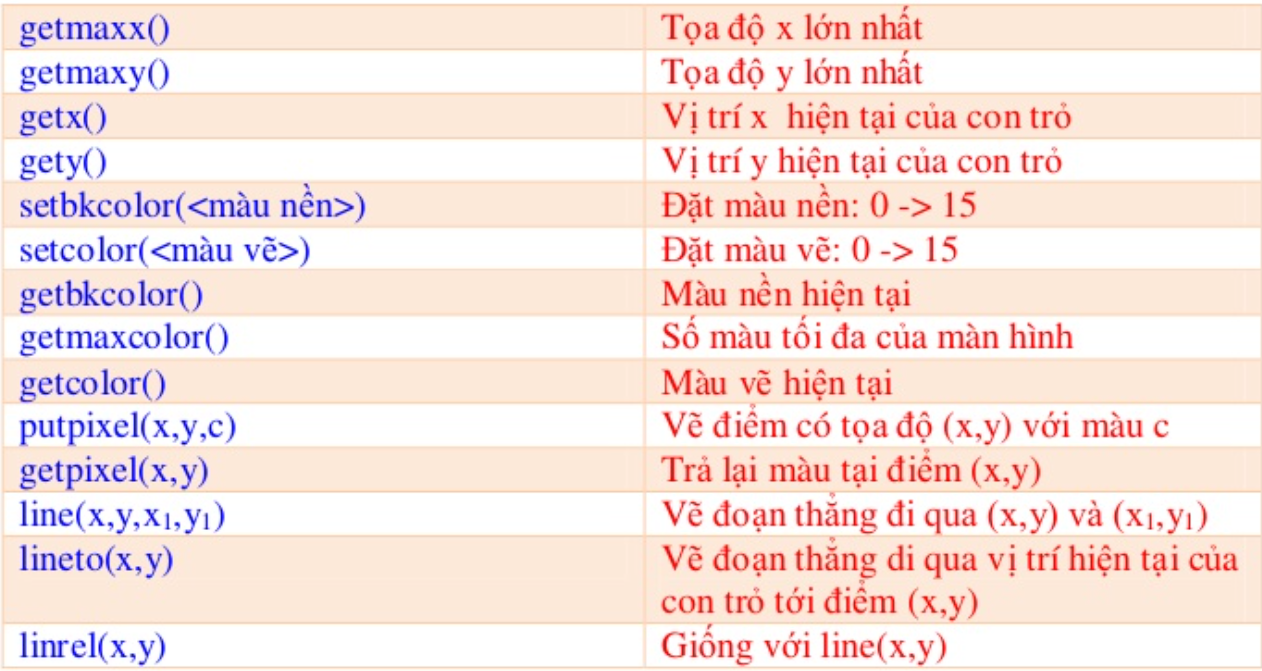
+Hàm trả lại 0 => không có lỗi gì, từ 1->18 tương ứng với các lỗi.

- Xoá tất cả những hình đã vẽ trên màn hình đồ hoạ:

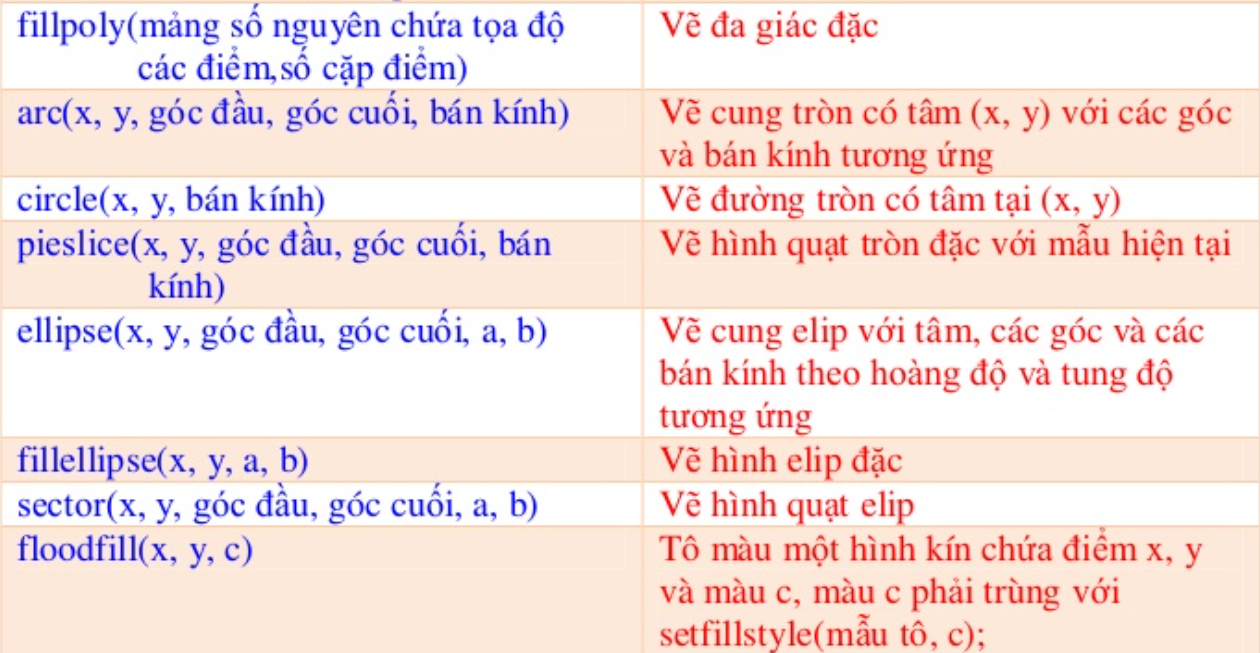
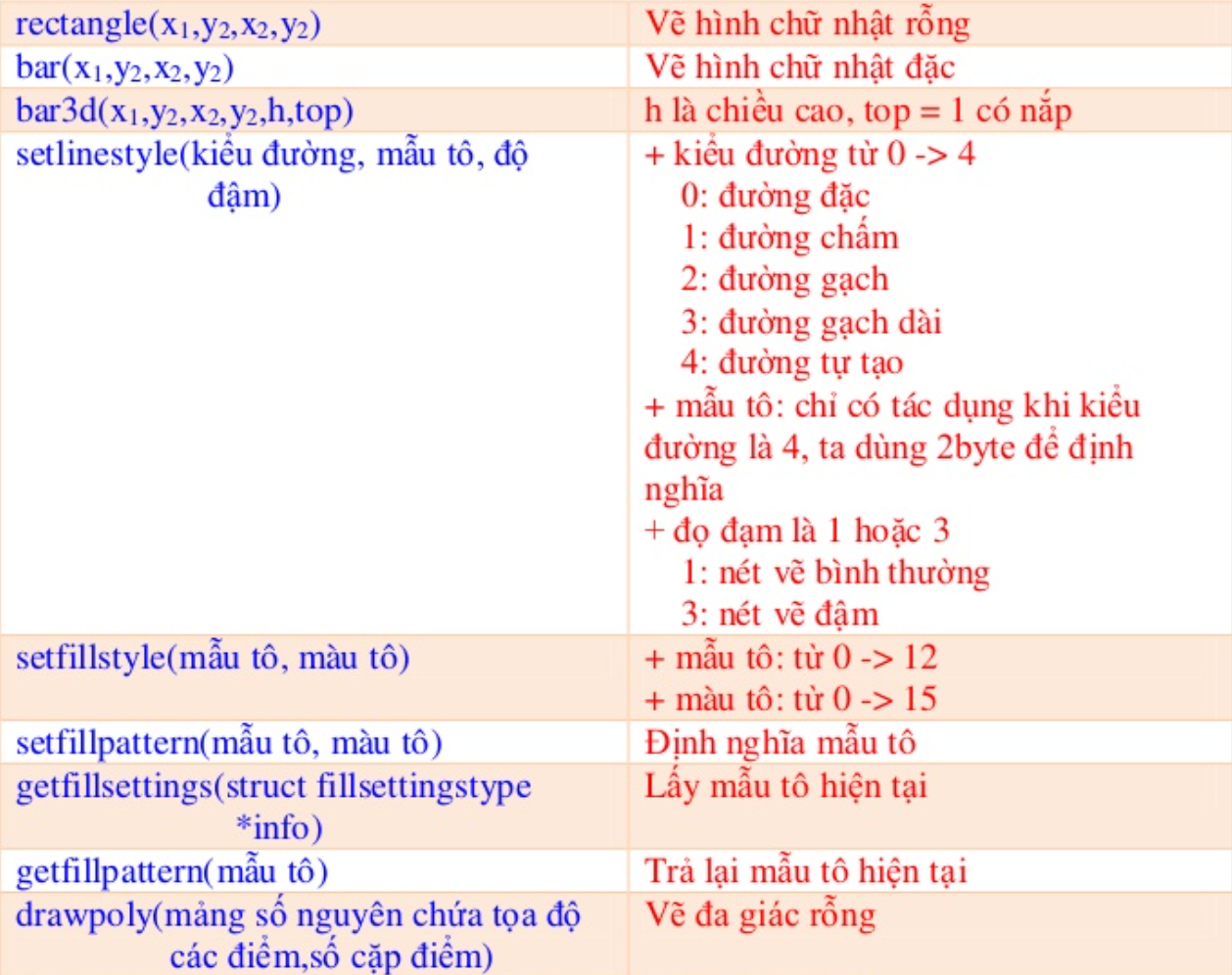
cleardevice ();

- Kết thúc chế độ đồ hoạ:

closegraph ();

- Các hàm thường dùng:

***Hình 7. Bảng các hàm thường dùng trong thư viện <graphics.h> (part 1).***



***Hình 8. Bảng các hàm thường dùng trong thư viện <graphics.h> (part 2).***

- Viết văn bản trong màn hình đồ hoạ:

outtextxy (x, y, s): viết văn bản tại vị trí (x, y).

moveto (x, y): chuyển con trỏ chuột đến vị trí (x, y).

- Điều chỉnh Font chữ, hướng chữ, cỡ chữ:

settextstyle (Font, hướng, cỡ chữ)

+Font:

DEFAULT\_FONT 0

SMALL\_FONT 1

TRIPLEX\_FONT 2

SANS\_SERIF\_FONT 3

GOTHIC\_FONT 4

+Hướng:

HOIRIZ\_DIR 0: nằm hàng ngang

VERT\_DIR 1: nằm hàng dọc

+Cỡ chữ bắt đầu từ 1.

- Điều chỉnh cách viết

Settextjustify (theo hướng ngang, theo hướng dọc)

+Theo hướng nằm ngang:

LEFT\_TEXT = 0: Viết từ trái sang phải

CENTER\_TEXT = 1: Viết từ vị trí con trỏ sang 2 bên

RIGHT\_TEXT = 2: Viết từ phải sang trái

+Theo hướng thẳng đứng:

BOTTOM\_TEXT = 0: Viết từ dưới lên

CENTER\_TEXT = 1: Viết từ vị trí con trỏ lên trên và xuống

TOP\_TEXT = 2: Viết từ trên xuống

* + - 1. Cách tính toạ độ trong đồ thị

**- Mô tả:**

**+** Các đỉnh của đồ thị sẽ biểu diễn bởi 1 hình tròn nhỏ nằm trên đường tròn.

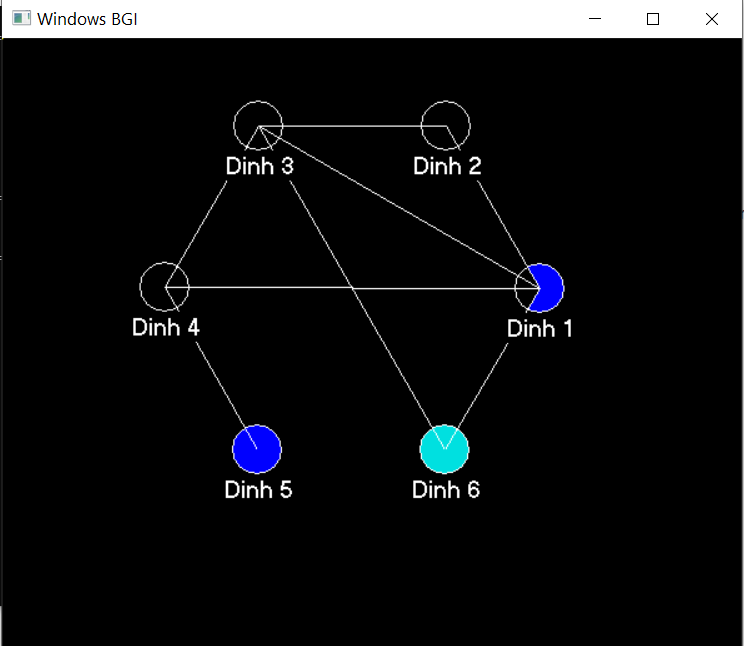
+ Các đỉnh của đồ thị sẽ cách đều nhau bằng cách chia hình tròn bằng nhiều phần bằng nhau

+ VD: Có 6 đỉnh thì sẽ chia hình tròn ra làm 6 phần bằng nhau tương ứng với các góc 0, 60, 120, 180, 240, 300 (trong chương trình sẽ được tính theo radian).

+ Sau đó các liên kết sẽ được thêm vào giữa các đỉnh có liên kết bằng hàm line().

\*\***Lưu ý:**

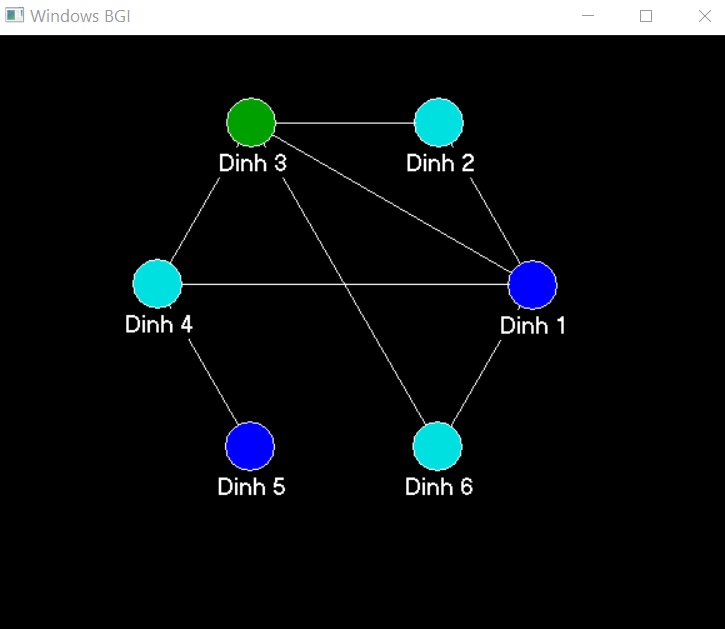
**+** Khi vẽ line cần chú ý cẩn thận vì có thể là khi vẽ các hình xuyên qua tâm của hình tròn tương ứng với đỉnh thì sẽ có thể làm border phần cần tô màu của đỉnh sẽ bị lệch đi khiến cho đỉnh không được tô theo như ý muốn.



***Hình 9. Console bị lỗi do các đoạn thẳng dính liền với nhau.***

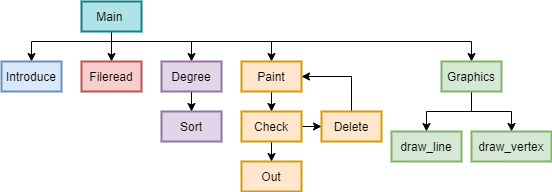
**+** Vì điểm được chọn để tô màu sẽ nằm trên đường line nên màu sẽ không được tô giống như đỉnh 3, đỉnh 2, đỉnh 4; hoặc là sẽ tô chỉ được 1 phần như là đỉnh 1.

+ Để khắc phục điều này thì sẽ cần cắt ngắn line đi chỉ đến phần viền của hình tròn (biểu diễn đỉnh) => Hàm draw\_line ().



***Hình 10. Console tô màu hoàn chỉnh (đã loại bỏ đoạn dư thừa của các đoạn thẳng).***

+ Hàm draw\_line () sẽ sử dụng các công thức sin, cos, tan để tính được đoạn dư thừa để cắt ngắn đi đoạn line dư ra.

* 1. Thuật toán

***Hình 11. Sơ đồ cấu trúc chương trình***

* Thuật toán:

1. Tính giá trị bậc của các đỉnh trong V. Lập danh sách V’ = [v1, v2, ..., vn] là các đỉnh của đồ thị được sắp xếp theo thứ tự bậc giảm dần: d(v1) >   
   d(v2) > ... > d(vn). Ban đầu tất cả các đỉnh trong V (V’) đều chưa được   
   tô màu.

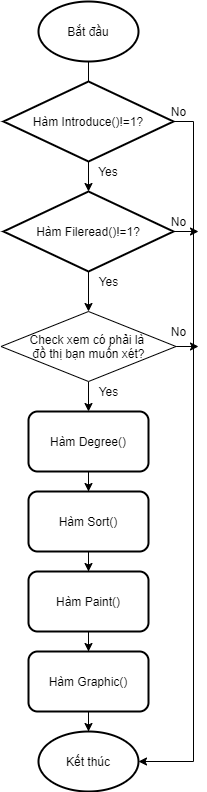
Gán color = 1.

1. Tô màu i cho đỉnh đầu tiên trong danh sách V’. Duyệt lần lượt các đỉnh khác trong U (nếu có) và chỉ tô màu color cho các đỉnh không kề đỉnh đã có màu color.
2. Kiểm tra nếu tất cả các đỉnh trong V đã được tô màu thì thuật toán kết thúc, đồ thị đã sử dụng color màu để tô. Ngược lại, nếu vẫn còn đỉnh chưa được tô thì chuyển sang Bước 4.
3. Loại khỏi danh sách V’ các đỉnh đã tô màu. Sắp xếp lại các đỉnh trong V’ theo thứ tự bậc giảm dần. Gán color = color + 1.

Quay lại Bước 2.

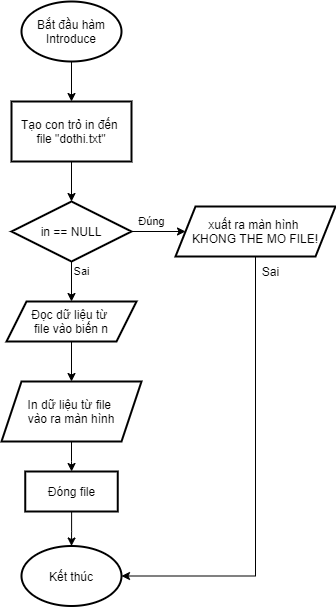
**Bước 5:** Vẽ đồ thị qua console thứ 2.

* **Lưu đồ thuật toán**:
* Hàm main



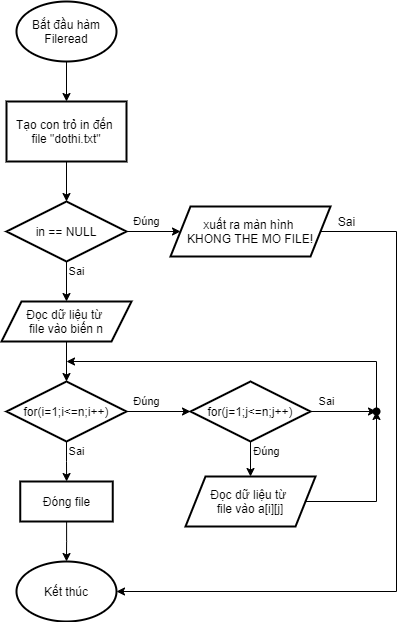
***Hình 12. Sơ đồ khối hàm main***

- Hàm Introduce:



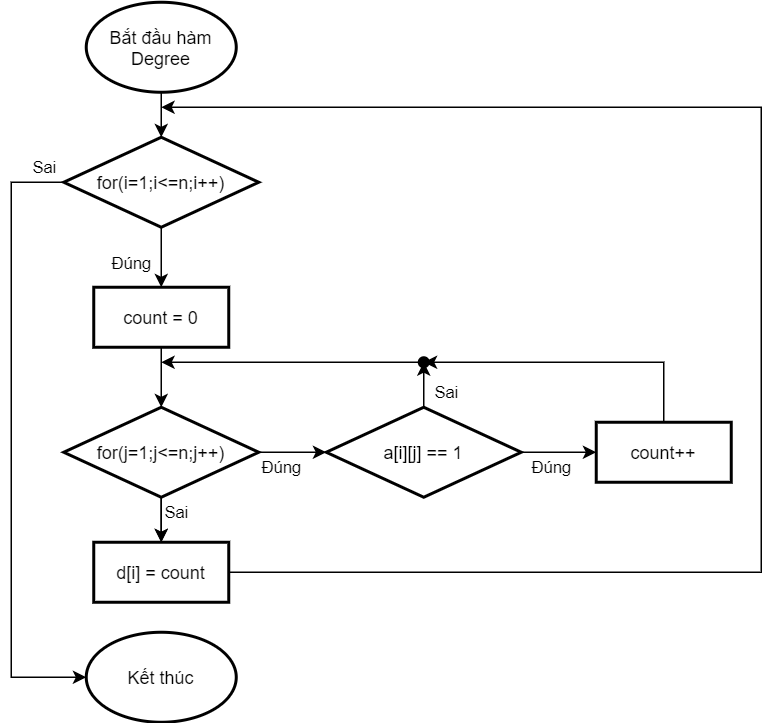
***Hình 13. Sơ đồ khối hàm Introduce.***

Hàm Fileread

******

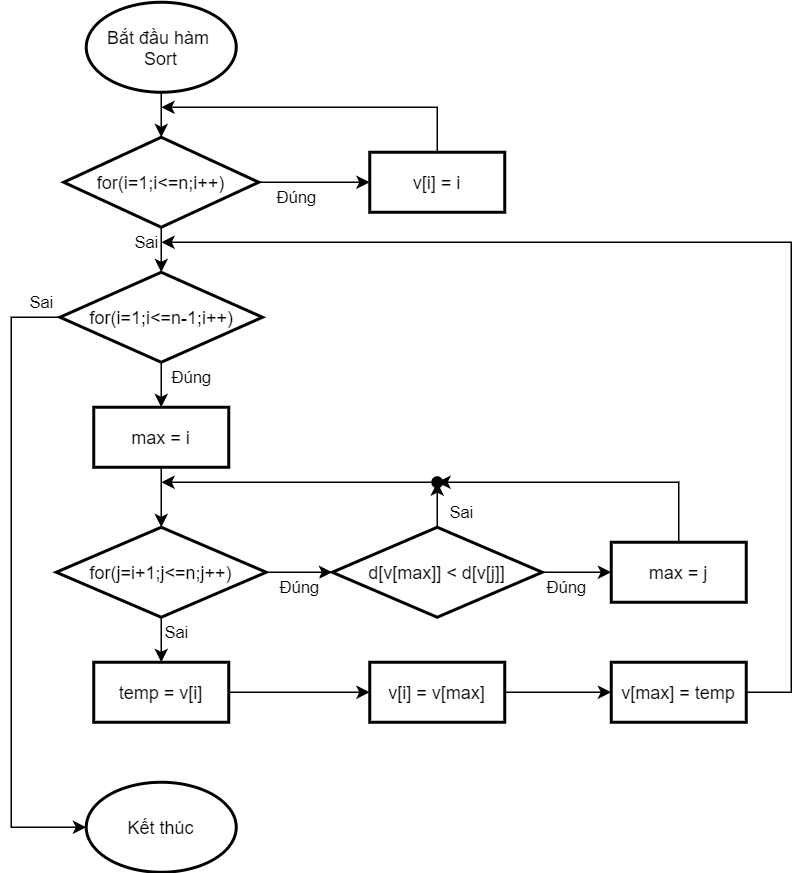
***Hình 14. Sơ đồ khối hàm Fileread***

* Hàm Degree

******

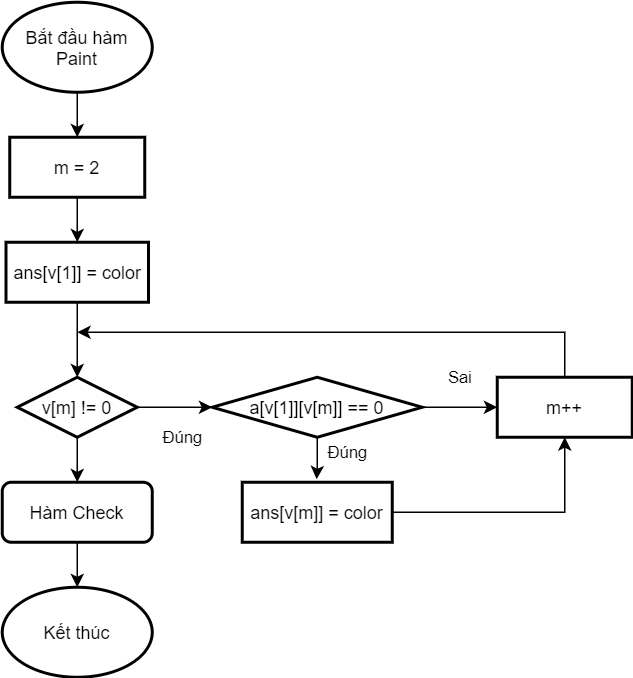
***Hình 15. Sơ đồ khối hàm Degree***

* Hàm Sort

******

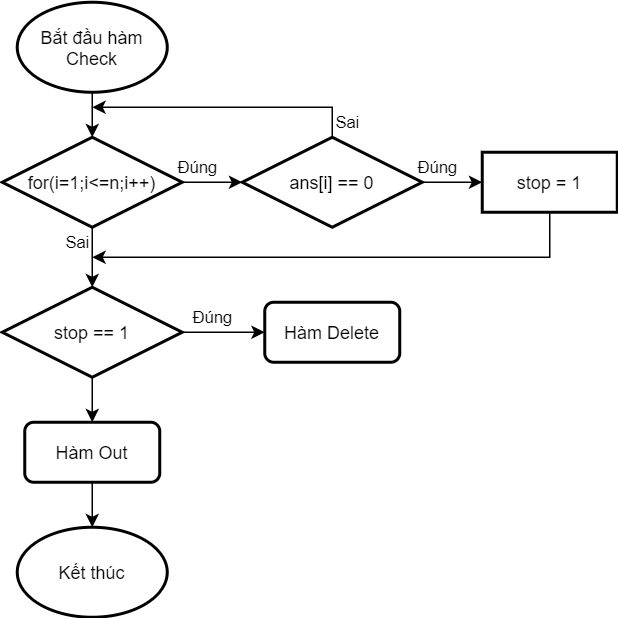
***Hình 16. Sơ đồ khối hàm Sort***

* Hàm Paint

******

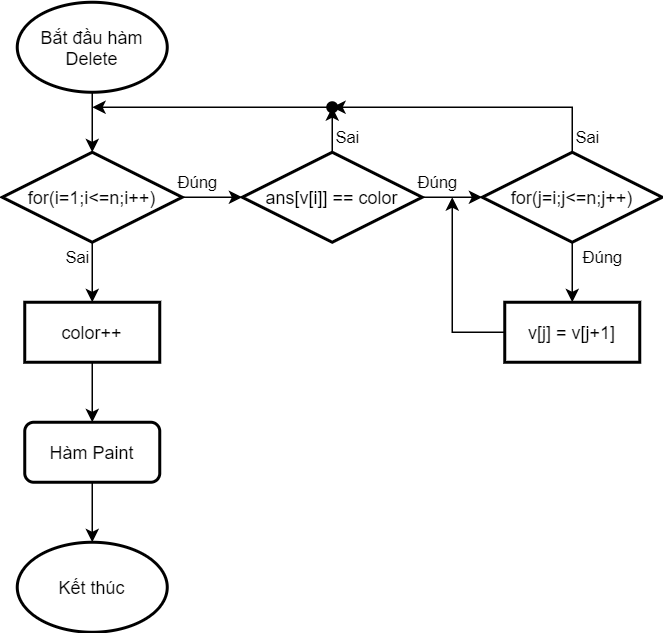
***Hình 17. Sơ đồ khối hàm Paint***

* Hàm Check



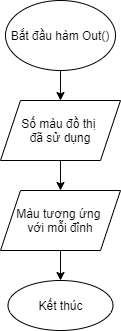
***Hình 18. Sơ đồ khối hàm Check***

* Hàm Delete

******

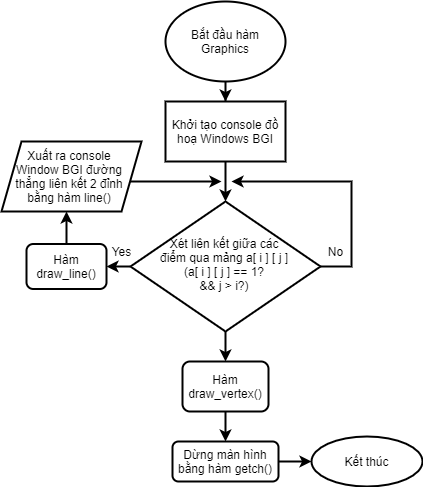
***Hình 19. Sơ đồ khối hàm Delete***

* Hàm Out:

******

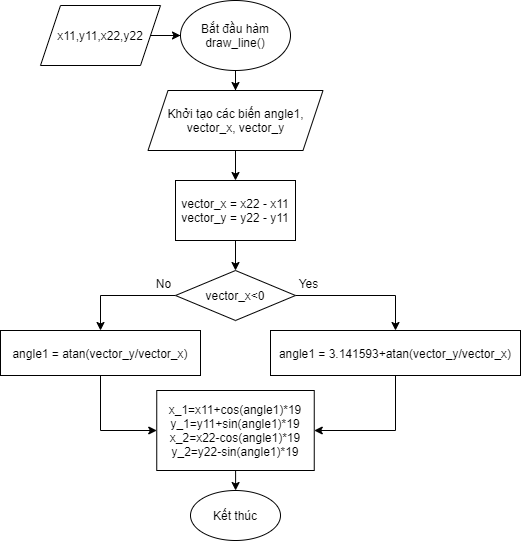
***Hình 20. Sơ đồ khối hàm Out***

* Hàm Graphics:

******

***Hình 21. Sơ đồ khối hàm Graphics***

* Hàm draw\_line:

******

***Hình 22. Sơ đồ khối hàm draw\_line***

* Hàm draw\_vertex:

Diagram

Description automatically generated

***Hình 23. Sơ đồ khối hàm draw\_vertex***

1. CHƯƠNG TRÌNH VÀ KẾT QUẢ
   1. Tổ chức chương trình

Xây dựng chương trình với các hàm:

- Hàm main: Hỏi xem đồ thị đưa vào đã đúng là đồ thị người sử dụng muốn xét không và gọi các hàm con.

* Hàm Introduce: Bắt đầu chương trình, in ra tên nhóm, tên đề tài, thành viên và giáo viên hướng dẫn.
* Hàm Fileread: Đọc dữ liệu từ file dothi.txt để gán vào biến n và mảng 2 chiều a.
* Hàm Degree: Tính giá trị bậc của đỉnh.
* Hàm Sort: Tạo mảng v chứa các đỉnh. Sắp xếp theo thứ tự giảm dần về bậc của đỉnh: d[v[1]] > d[v[2]] > … > d[v[n]].
* Hàm Paint: Tô màu color cho đỉnh đầu tiên của mảng v (v[1]) và các đỉnh không kề đỉnh v[1] (a[v[1]][v[m]] == 0).
* Hàm Check: Kiểm tra tất cả các đỉnh đã được tô màu hay chưa. Nếu đã tô hết thì chuyển sang hàm Out. Còn không thì chuyển sang hàm Delete.
* Hàm Delete:
* Xóa đỉnh đã được tô màu color trong mảng v;
* Tăng giá trị cho biến color: color = color + 1;
* Quay lại hàm Paint.
* Hàm Out: Xuất ra số màu đã dùng, đỉnh và màu được tô ứng với đỉnh đó.
* Hàm Graphics: Gọi các hàm draw\_line (vẽ đường thẳng) và draw\_vertex (vẽ đường thẳng biểu diễn đỉnh).
  1. Ngôn ngữ cài đặt

Sử dụng ngôn ngữ lập trình C.

* + 1. Giới thiệu

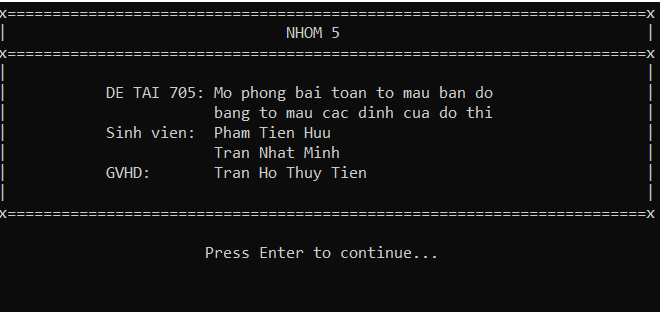
Ngôn ngữ lập trình C là một ngôn ngữ mệnh lệnh được phát triển từ đầu thậpniên 1970 bởi Dennis Ritchie để dùng trong hệ điều hành UNIX. Từ đó, ngôn ngữ này đã lan rộng ra nhiều hệ điều hành khác và trở thành một những ngôn ngữ phổ dụng  
nhất. C là ngôn ngữ rất có hiệu quả và được ưa chuộng nhất để viết các phần mềm hệ  
thống, mặc dù nó cũng được dùng cho việc viết các ứng dụng. Ngoài ra, C cũng  
thường được dùng làm phương tiện giảng dạy trong khoa học máy tính mặc dù ngôn  
ngữ này không được thiết kế dành cho người nhập môn.

* + 1. Tổng quan

C là một ngôn ngữ lập trình tương đối nhỏ gọn vận hành gần với phần cứng và nó giống với ngôn ngữ Assembler hơn hầu hết các ngôn ngữ bậc cao. Hơn thế, C đôi khi được đánh giá như là "có khả năng di động", cho thấy sự khác nhau quan trọng  
giữa nó với ngôn ngữ bậc thấp như là Assembler, đó là việc mã C có thể được dịch và  
thi hành trong hầu hết các máy tính, hơn hẳn các ngôn ngữ hiện tại trong khi đó  
thì Assembler chỉ có thể chạy trong một số máy tính đặc biệt. Vì lý do này C được  
xem là ngôn ngữ bậc trung.

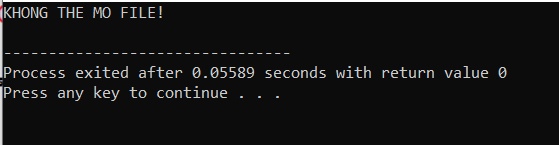
C đã được tạo ra với một mục tiêu là làm cho nó thuận tiện để viết các chương  
trình lớn với số lỗi ít hơn trong mẫu hình lập trình thủ tục mà lại không đặt gánh nặng  
lên vai người viết ra trình dịch C, là những người bề bộn với các đặc tả phức tạp của  
ngôn ngữ. Cuối cùng C có thêm những chức năng sau:

* Một ngôn ngữ cốt lõi đơn giản, với các chức năng quan trọng chẳng hạn như là  
  những hàm hay việc xử lý tập tin sẽ được cung cấp bởi các bộ thư viện các thủ tục.
* Tập trung trên mẫu hình lập trình thủ tục, với các phương tiện lập trình theo  
  kiểu cấu trúc.
* Một hệ thống kiểu đơn giản nhằm loại bỏ nhiều phép toán không có ý nghĩa thực dụng.
* Dùng ngôn ngữ tiền xử lý, tức là các câu lệnh tiền xử lý C, cho các nhiệm vụ như là định nghĩa các macro và hàm chứa nhiều tập tin mã nguồn (bằng cách dùng câu lệnh tiền xử lý dạng #include chẳng hạn).
* Mức thấp của ngôn ngữ cho phép dùng tới bộ nhớ máy tính qua việc sử dụng kiểu dữ liệu pointer.
* Số lượng từ khóa rất nhỏ gọn.
* Các tham số được đưa vào các hàm bằng giá trị, không bằng địa chỉ.
* Hàm các con trỏ cho phép hình thành một nền tảng ban đầu cho tính đóng và tính đa hình.
* Hỗ trợ các bản ghi hay các kiểu dữ liệu kết hợp do người dùng từ khóa định  
  nghĩa struct cho phép các dữ liệu liên hệ nhau có thể được tập hợp lại và được  
  điều chỉnh như là toàn bộ.
  1. Kết quả
     1. Giao diện chính của chương trình

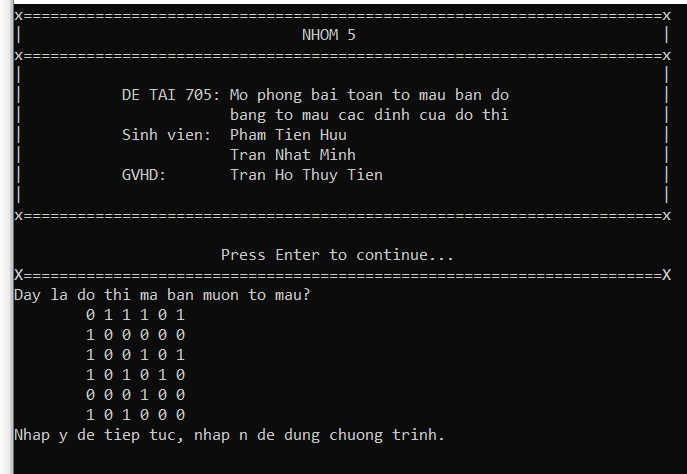
****- Ngay khi vừa thực thi chương trình:**

***Hình 24. Giao diện chính của chương trình lúc ban đầu***

\*\*Trong quá trình nhập liệu có thể xảy ra trường hợp file “Introduce.txt” bị xoá hoặc bị di chuyển đi thư mục khác. Nếu không có file “Introduce.txt” ngay trong thư mục chứa chương trình thì chương trình sẽ hiển thị như sau:

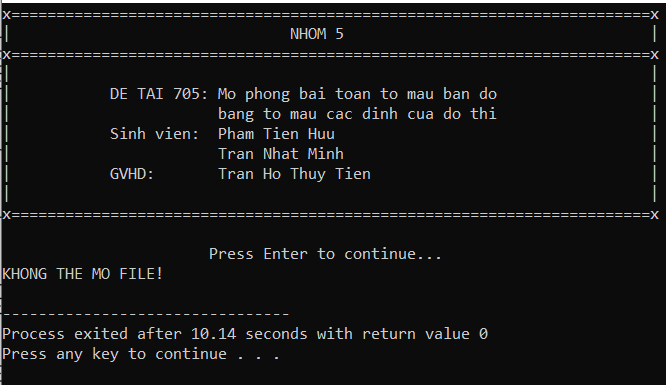


***Hình 25. Không tìm thấy file “Introduce.txt”***

**- Sau khi nhấn Enter:**

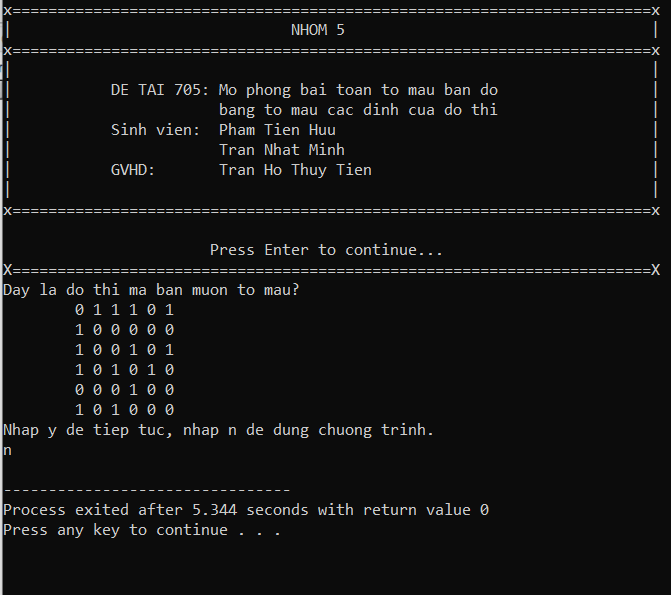
***Hình 26. Giao diện của chương trình sau khi nhấn Enter***

***(Kiểm tra lại dữ liệu đầu vào trước khi thực thi)***

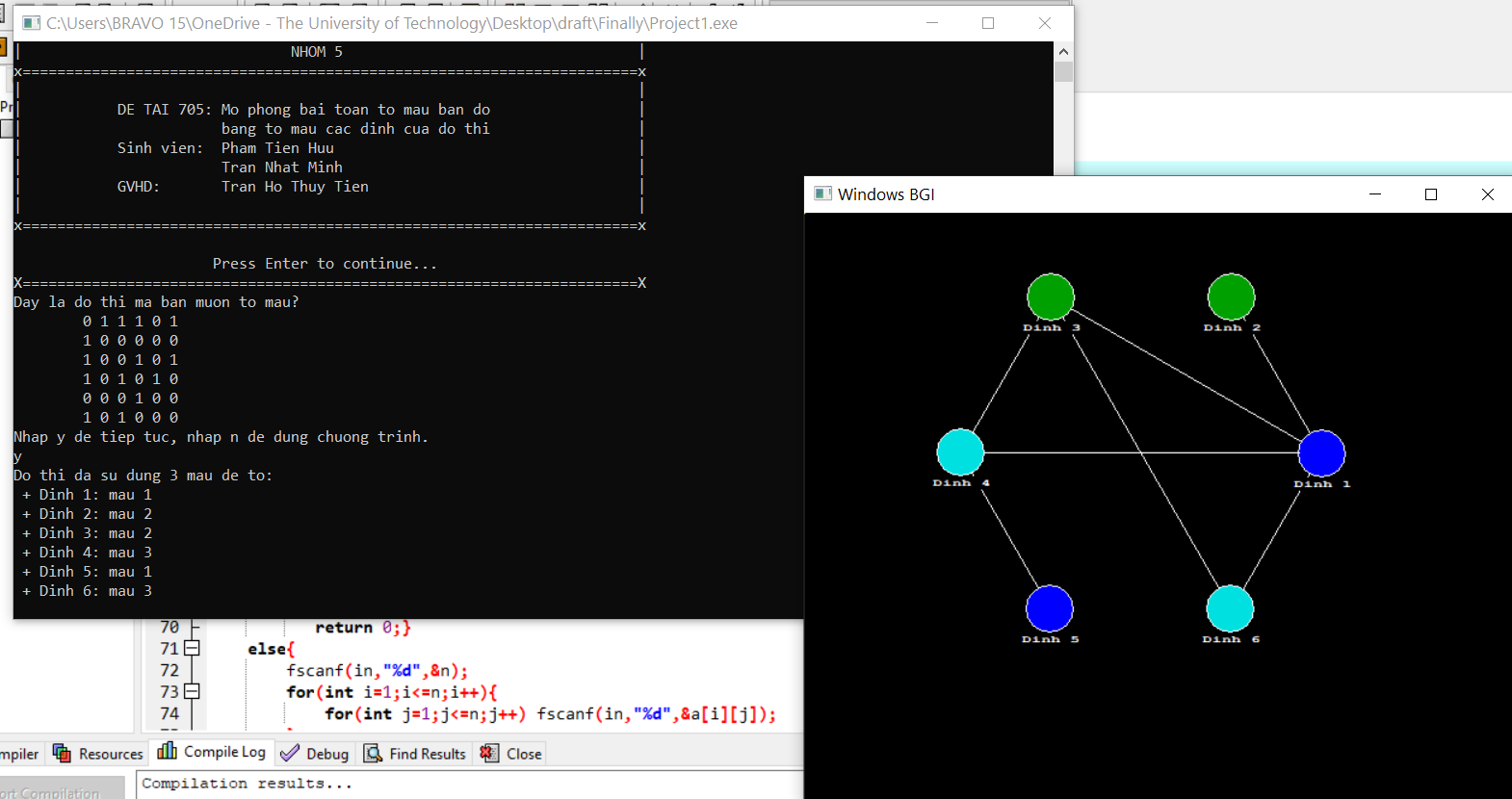
******

***Hình 27. Không tìm thấy file “dothi.txt”***

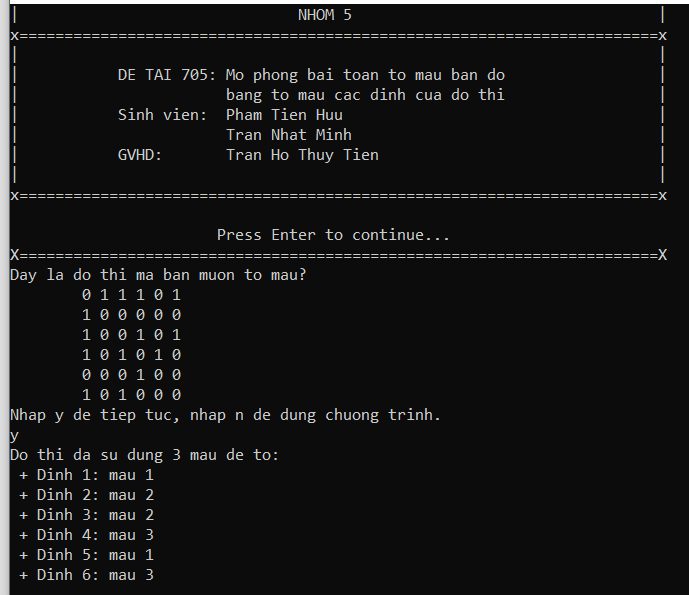
\*\*Nếu cảm thấy đồ thị truyền vào chưa đúng như ý muốn, bạn có thể nhập “n” và nhấn Enter nếu muốn dừng chương trình tại ngay bước này:

******

***Hình 28. Dừng file khi thấy đồ thị nhập vào chưa đúng***

* + 1. Kết quả thực thi của chương trình

***Hình 29. Kết quả thu được (toàn màn hình)***

******

***Hình 30. Console 1 (kết quả tô màu)***

****

***Hình 31. Console 2 (phác thảo đồ thị)***

* 1. Nhận xét đánh giá

Lúc ban đầu nhận đề tài, chúng em cảm thấy có rất nhiều vấn đề khó, nhiều vấn đề cần phải giải quyết, nhưng sau quá trình tìm hiểu, nghiên cứu, trải qua rất nhiều các phiên bản để thử, được giáo viên hướng dẫn, tận tình chỉ rõ những vấn đề, chúng em dần rút ra kinh nghiệm, tìm ra hướng giải quyết, gỡ các nút thắt vấn đề và dần dần đi đến hoàn thiện chương trình và đáp ứng yêu cầu và mục tiêu của đề tài.

Nhìn chung chương trình của chúng em đã giải quyết được vấn đề về tô màu đồ thị nhưng mới ở mức căn bản và chưa chuyên sâu. Chúng em cần tìm hiểu thêm và cải thiện chương trình và ứng dụng ngày càng trở nên khoa học, đáp ứng được nhiều ứng dụng của lý thuyết đồ thị hơn.

So với việc sử dụng kỹ thuật đệ quy quay lui, tốc độ xử lý khi sử dụng tô màu đồ thị cao hơn nhiều. Tuy nhiên, một điều ta cần phải lưu ý rằng: Thuật toán tô màu đồ thị được nêu ở trên, dù được sử dụng rất rộng rãi, nhưng đã được chứng minh là không đúng trong mọi trường hợp. Thuật toán chỉ đảm bảo việc tô màu sẽ thỏa mãn các rảng buộc, nhưng tiêu chí số màu tô là tối thiểu thì có thể không đạt được. Vì thế các nhà khoa học hiện tại vẫn coi thuật toán này đúng một cách xấp xỉ (tuyệt đại đa số trường hợp). Cũng chính vì vậy, thay vì việc phải đi tìm chính xác sắc số của một đồ thị nào đó, người ta thường tiếp cận vấn đề theo hướng trả lời cho câu hỏi: Có thể dùng m màu để tô cho một đồ thị cụ thể có n đỉnh hay không? Trong trường hợp cụ thể của bài toán Sudoku là trả lời câu hỏi liệu có thể dùng 9 màu để tô cho đồ thị 81 đỉnh hay không? Thuật toán tô màu trên có thể được sử dụng để tô màu đồ thị phẳng và không phẳng. Đối với đồ thị phẳng thì đây là một bài toán kinh điển, và sắc số của một đồ thị phẳng đã được chứng minh là không quá 4 (định lý 4 màu), đối với đồ thị không phẳng, sắc số sẽ lớn hơn.

Vấn đề liên quan đến tô màu đồ thị nói chung hiện vẫn chưa có được những thống nhất trọn vẹn, nhất là về tính đúng của sắc số trong mọi trường hợp hay độ phức tạp tính toán đa thức hay phi đa thức của thuật toán. Tuy không đảm bảo tính tối ưu, nhưng việc đảm bảo về ràng buộc, thuật toán tô màu đồ thị nói trên đã giúp chúng ta giải quyết rất nhiều bài toán quan trọng được đặt ra trong thực tiễn.

1. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN
   1. Kết luận

Sau khi tìm hiểu và thực hiện đề tài: “Mô tả bài toán tô màu bản đồ bằng tô màu các đỉnh của đồ thị” em đã thực hiện được đề tài theo yêu cầu cơ bản ban đầu. Trong quá trình thực hiện đề tài em đã hiểu hơn về sắp xếp mảng, các thao tác với file, các thư viện đồ hoạ trong C, biết cách tổ chức chương trình hợp lý. Về kết quả chương trình đáp ứng được những yêu cầu đặt ra.

* 1. Hướng phát triển

Trong phạm vi đồ án chúng em chỉ trình bày cách để tô màu 1 đồ thị phẳng chứ chưa phân biệt được dữ liệu đầu vào có phải là đồ thị phẳng hay không. Tuy nhiên chúng ta có thể cải tiến chương trình sử dụng nhiều thuật toán hơn để phân biệt dữ liệu đầu vào có phải là đồ thị phẳng hay không, từ đó ta có thể xem xét tính tối ưu của thuật toán tô màu đồ thị trên có thoả mãn định lý 4 màu hay không.

Ngoài ra, bài toán trên có rất nhiều ứng dụng trong thực tế, ví dụ: xây dựng mạch điện, xác định hai máy tính có kết nối hay không, xác định đường đi ngắn nhất giữa hai thành phố, phân chia kênh truyền cho đài truyền hình, lập lịch thi. Có thể biến đổi chương trình nhưng vẫn sử dụng các hàm cũ để giải quyết các vấn đề trên.

TÀI LIỆU THAM KHẢO

1. **8 kiểu cấu trúc dữ liệu mà mọi lập trình viên cần phải biết**

[*https://viblo.asia/p/8-kieu-cau-truc-du-lieu-ma-moi-lap-trinh-vien-can-phai-biet-yMnKMmdEK7P*](https://viblo.asia/p/8-kieu-cau-truc-du-lieu-ma-moi-lap-trinh-vien-can-phai-biet-yMnKMmdEK7P)

1. **Đọc – ghi file trong C**

*https://freetuts.net/doc-ghi-file-trong-c-2991.html*

1. **Thuật toán sắp xếp selection sort**

[*https://nguyenvanhieu.vn/thuat-toan-sap-xep-selection-sort/*](https://nguyenvanhieu.vn/thuat-toan-sap-xep-selection-sort/)

1. **C (ngôn ngữ lập trình)**

[*https://vi.wikipedia.org/wiki/C\_(ng%C3%B4n\_ng%E1%BB%AF\_l%E1%BA%ADp\_tr%C3%ACnh)/*](https://vi.wikipedia.org/wiki/C_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)/)

**5. Tô màu đồ thị**

*https://vi.wikipedia.org/wiki/T%C3%B4\_m%C3%A0u\_%C4%91%E1%BB%93\_th%E1%BB%8B*

**6. Bảng màu**

*https://www.geeksforgeeks.org/setcolor-function-c/*

**7. Lý thuyết đồ hoạ trong C**

*https://www.slideshare.net/ANHMATTROI/mot-so-ham-do-hoa-trong-c-c*

PHỤ LỤC

* **Chương trình:**

#include <stdio.h>

#include <graphics.h>

#include <conio.h>

#include <math.h>

int a[20][20],v[20],d[20],ans[20],n,color=1,radius = 150;

double angle,x\_1,x\_2,y\_1,y\_2;

int Introduce();

int Fileread();

void Degree();

void Sort();

void Paint();

void Delete();

void Check();

void Out();

void Graphics();

void draw\_line(double x11, double y11, double x22, double y22);

void draw\_vertex();

int main(){

if (Introduce()!=1) return 0;

if (Fileread()!=1) return 0;

char sign;

printf("Day la do thi ma ban muon to mau?\n");

for(int i=1;i<=n;i++){

for(int j=1;j<=n;j++){

printf("%d ",a[i][j]);

}

printf("\n");

}

printf("Nhap y de tiep tuc, nhap n de dung chuong trinh.\n");

while (sign!='y' && sign!='n') {

sign = getchar();

}

if (sign=='n') return 0;

Degree();

Sort();

Paint();

Graphics();

return 0;

}

int Introduce(){

FILE \*in;

in=fopen("Introduce.txt","r");

char c;

if(in==NULL){

printf("KHONG THE MO FILE!\n");

return 0;}

else{

do{

c=fgetc(in);

printf("%c",c);

}

while(c!=EOF);

fclose(in);

}

getchar();

return 1;

}

int Fileread(){ //lay du lieu tu file de tao mang a[][]

FILE \*in;

in=fopen("dothi.txt","r");

if(in==NULL){

printf("KHONG THE MO FILE!\n");

return 0;}

else{

fscanf(in,"%d",&n);

for(int i=1;i<=n;i++){

for(int j=1;j<=n;j++) fscanf(in,"%d",&a[i][j]);

}

fclose(in);

}

return 1;

}

void Degree(){

for(int i=1;i<=n;i++){ //xet ma tran a[][] neu hang i co "count" so 1 -> bac cua dinh i la "count"

int count=0;

for(int j=1;j<=n;j++){

if(a[i][j]==1) count++;

}

d[i]=count;

}

}

void Sort(){ //sap xep mang dinh v[] theo thu tu giam dan ve bac

for(int i=1;i<=n;i++){ //tao mang dinh

v[i]=i;

}

for(int i=1;i<=n-1;i++){

int max=i;

for(int j=i+1;j<=n;j++){

if(d[v[max]]<d[v[j]]) max=j;

}

int temp=v[i];

v[i]=v[max];

v[max]=temp;

}

}

void Paint(){

int m=2;

ans[v[1]]=color;

while(v[m]){

if(a[v[1]][v[m]]==0){ //neu khong ke dinh thi to mau

ans[v[m]]=color;

}

m++;

}

Check();

}

void Check(){

bool stop;

for(int i=1;i<=n;i++){

if(ans[i]==0){

stop=1;

break;

}

}

if(stop==1) Delete();

else Out();

}

void Delete(){

for(int i=1;i<=n;i++){

if(ans[v[i]]==color){

for(int j=i;j<=n;j++){

v[j]=v[j+1];

}

}

}

color++;

Paint();

}

void Out(){

printf("Do thi da su dung %d mau de to:\n",color);

for(int i=1;i<=n;i++){

printf(" + Dinh %d: mau %d\n",i,ans[i]);

}

}

void Graphics(){

int i,j;

initwindow(600,500);

angle = 6.283185/n;

for(i=1;i<=n;i++){

for(j=1;j<=n;j++){

if (a[i][j]==1 && j>i){

draw\_line(280+cos(angle\*(i-1))\*radius, 200-sin(angle\*(i-1 ))\*radius,280+cos(angle\*(j-1))\*radius, 200-sin(angle\*(j-1 ))\*radius);

line(x\_1,y\_1,x\_2,y\_2);

}

}

}

draw\_vertex();

getch();

}

void draw\_line(double x11, double y11, double x22, double y22){

double angel, vector\_x, vector\_y;

vector\_x=x22-x11;

vector\_y=y22-y11;

if (vector\_x<0) angel = 3.141593+atan(vector\_y/vector\_x);

else angel = atan(vector\_y/vector\_x);

x\_1=x11+cos(angel)\*19;

y\_1=y11+sin(angel)\*19;

x\_2=x22-cos(angel)\*19;

y\_2=y22-sin(angel)\*19;

}

void draw\_vertex(){

for(int i=1;i<=n;i++){

setfillstyle(USER\_FILL, ans[i]);

circle(280+cos(angle\*(i-1))\*radius, 200-sin(angle\*(i-1))\*radius, 20);

floodfill(280+cos(angle\*(i-1))\*radius+1, 200-sin(angle\*(i-1))\*radius+1,15);

settextstyle(3,0,1);

char vertexnum[10];

sprintf(vertexnum,"Dinh %d",i);

outtextxy(280+cos(angle\*(i-1))\*radius-26, 200-sin(angle\*(i- 1))\*radius+20,vertexnum);

}

}