BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO THỰC TẬP CƠ SỞ**

**SỬ DỤNG THƯ VIỆN GRAPHICS.H**

**MÔ PHỎNG QUÁ TRÌNH SẮP XẾP MẢNG BẰNG GIẢI THUẬT QUICK-SORT**

**Giảng viên hướng dẫn: ThS. Đoàn Vũ Thịnh**

**Sinh viên thực hiện: Nguyễn Ngọc Thảo**

**Mã số sinh viên: 62139013**

KHÁNH HÒA-2023

TRƯỜNG ĐẠI HỌC NHA TRANG

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN HỆ THỐNG THÔNG TIN**



**BÁO CÁO THỰC TẬP CƠ SỞ**

**SỬ DỤNG THƯ VIỆN GRAPHICS.H**

**MÔ PHỎNG QUÁ TRÌNH SẮP XẾP MẢNG BẰNG GIẢI THUẬT QUICK-SORT**

Giảng viên hướng dẫn: ThS. Đoàn Vũ Thịnh

Sinh viên thực hiện: Nguyễn Ngọc Thảo

Mã số sinh viên: 62139013

Khánh Hòa, tháng 01/2023

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**Khoa: Công nghệ Thông tin**

**PHIẾU THEO DÕI TIẾN ĐỘ VÀ ĐÁNH GIÁ BÁO CÁO THỰC TẬP CƠ SỞ**

**Tên đề tài:** SỬ DỤNG THƯ VIỆN GRAPHICS.H MÔ PHỎNG QUÁ TRÌNH SẮP XẾP MẢNG BẰNG GIẢI THUẬT QUICK-SORT

**Giảng viên hướng dẫn:** ThS. Đoàn Vũ Thịnh

**Sinh viên được hướng dẫn:** Nguyễn Ngọc Thảo

**MSSV:** 62139013

**Khóa:** 62 **Ngành:** Công nghệ Thông tin

|  |  |  |  |
| --- | --- | --- | --- |
| **Lần** | **Ngày** | **Nội dung** | **Nhận xét của GVHD** |
| 1 | 5/12/2022 | Nhận đề tài hướng dẫn và định hướng giải quyết vấn đề. Sinh viên trình bày kế hoạch thực hiện. |  |
| 2 | 11/12/2022 | Sinh viên trình bày việc mô phỏng thuật toán chính dựa trên kiến thức đã được học ở môn kỹ thuật đồ họa và các kiến thức thu nhận được từ Internet để minh họa bài toán đa dạng nhất có thể. |  |
| 3 | 23/12/2022 | Sinh viên hoàn thiện các thuật toán đã đề ra. Trình bày thuật toán với các trường hợp sai và chỉ ra được hướng khắc phục cho các trường hợp đó, thêm màu khi sắp xếp nhưng còn lỗi màu. |  |
| 4 | 05/01/2021 | Sinh viên nộp bản thảo của báo cáo thực tập lần thứ 1 và tiến hành chỉnh sửa. |  |

**Nhận xét chung (sau khi sinh viên hoàn thành ĐA/KL)**:

Điểm hình thức:

Điểm nội dung:

**Điểm tổng kết:**

Đồng ý cho sinh viên: Được bảo vệ: 🗹 Không được bảo vệ: 

Khánh Hòa, ngày 20 tháng 01 năm 2021

Cán bộ hướng dẫn

*(Ký và ghi rõ họ tên)*

# LỜI CẢM ƠN

Để có thể hoàn thành đợt thực tập lần này, chúng em xin chân thành cảm ơn đến quý thầy cô khoa Công nghệ Thông tin đã tạo điều kiện hỗ trợ và giúp đỡ chúng em trong quá trình học tập và nghiên cứu đề tài này.

Qua đây, nhóm xin chân thành cảm ơn thầy Đoàn Vũ Thịnh, người đã trực tiếp quan tâm và hướng dẫn chúng em hoàn thành tốt đợt thực tập trong thời gian qua.

Do kiến thức còn hạn chế và thời gian thực hiện còn ngắn nên bài báo cáo của chúng em còn nhiều thiếu sót, kính mong sự góp ý của quý thầy cô.

# MỤC LỤC

[LỜI CẢM ƠN i](#_Toc123567860)

[MỤC LỤC ii](#_Toc123567861)

[DANH MỤC HÌNH iii](#_Toc123567862)

[TÓM TẮT iv](#_Toc123567863)

[GIỚI THIỆU 1](#_Toc123567864)

[1.1. Dev C++ 2](#_Toc123567865)

[1.2. Thư viện Graphics.h 2](#_Toc123567866)

[1.3. Thuật toán sắp xếp. 3](#_Toc123567867)

[1.3.1. Một số thuật toán sắp xếp. 3](#_Toc123567868)

[1.4. Thuật toán sắp xếp Quick-Sort 3](#_Toc123567869)

[1.4.1. Sơ lược về thuật toán 3](#_Toc123567870)

[1.4.2. Ý tưởng chính của thuật toán 4](#_Toc123567871)

[1.4.3. Ví dụ minh họa thuật toán 4](#_Toc123567872)

[1.4.4. Độ phức tạp giải thuật 10](#_Toc123567873)

[1.4.5. Lưu đồ giải thuật 12](#_Toc123567874)

[CHƯƠNG 2. PHƯƠNG PHÁP NGHIÊN CỨU 13](#_Toc123567875)

[2.1. Cài đặt Dev C và thư viện graphics.h 13](#_Toc123567876)

[2.2. Cài đặt thuật toán tô màu 13](#_Toc123567877)

[2.3. Cài đặt thuật toán Quick-Sort 15](#_Toc123567878)

[2.3. Thực hiện phép tịnh tiến 16](#_Toc123567879)

[2.3. Hiển thị kết quả 20](#_Toc123567880)

[CHƯƠNG 3. KẾT QUẢ THỰC HIỆN 21](#_Toc123567881)

[3.1. Mô phỏng thuật toán 21](#_Toc123567882)

[3.1. Kết thúc mô phỏng 22](#_Toc123567883)

[THẢO LUẬN 23](#_Toc123567884)

[TÀI LIỆU THAM KHẢO 24](#_Toc123567885)

# DANH MỤC HÌNH

[*Hình 1.1. Ví dụ về đồ họa máy tính* 1](#_Toc17920)

[*Hình 1.2. Lưu đồ giải thuật Quick – Sort* 1](#_Toc14280)2

[*Hình 1.3. Ví dụ minh hoạ thư viện winbgim* 11](#_Toc14280)

[*Hình 3.1. Kết quả sau khi sắp xếp* 11](#_Toc14280)

# **TÓM TẮT**

Đồ họa máy tính là một lĩnh vực của Công nghệ Thông tin, ở đó việc nghiên cứu xây dựng và tập hợp các công cụ (mô hình lý thuyết và phần mềm) khác nhau để kiến tạo, xây dựng, lưu trữ và xử lý các mô hình và hình ảnh của các đối tượng, sự vật, hiện tượng trong cuộc sống, sản xuất. Thuật toán tô màu bằng dòng quét và tô màu bằng đường biên trong đồ họa máy tính có tầm quan trọng rất lớn và được sử dụng rộng rãi trong các phần mềm đồ họa phổ biến hiện nay như Adobe Photoshop, Corel Draw, Microsoft Paint.

Quy trình thực hiện được trải qua các bước từ cài đặt thuật toán, hiển thị kết quả đầu ra trên màn hình đều được thực hiện trên môi trường C/C++ thông qua ứng dụng Dev C/C++ có kết hợp với thư viện graphics.h.

Sản phẩm đã minh họa được từng bước giải thuật sắp xếp mảng Quick-Sort. Đồng thời cũng chỉ ra các trường hợp hạn chế của mỗi thuật toán và cách khắc phục các nhược điểm đó. Sản phẩm chạy tốt với dữ liệu được nhập từ bàn phím. Về yêu cầu sử dụng dữ liệu được nhập từ bàn phím để thực hiện sắp xếp với giải thuật Quick-Sort chưa được triển khai. Đây cũng là thiếu sót của đề tài ngay từ khi đặt ra.

Toàn bộ mã nguồn của báo cáo được tải lên theo địa chỉ:

*https://github.com/thinhdoanvu/DOAN\_TTCS\_TTCN/tree/main/NguyenNgocThao\_62139013*

# GIỚI THIỆU

Đồ họa máy tính là một lĩnh vực của Công nghệ thông tin, ở đó nghiên cứu, xây dựng và tập hợp các công cụ (mô hình lý thuyết và phần mềm) khác nhau để kiến tạo, xây dựng, lưu trữ và xử lý các mô hình và hình ảnh của các đối tượng, sự vật, hiện tượng trong cuộc sống, sản xuất, nghiên cứu. Đồ họa máy tính góp phần quan trọng làm cho giao tiếp giữa con người và máy tính trở nên thân thiện hơn. Từ đồ họa trên máy tính chúng ta có nhiều lĩnh vực có ứng dụng rất quan trọng của đồ họa máy tính trong thực tế như: tạo mô hình, hoạt cảnh, hỗ trợ thiết kế đồ họa, mô phỏng hình ảnh, chẩn đoán hình ảnh (trong Y tế), huấn luyện đào tạo ảnh (quân sự, hàng không…).



Hình 1.1. Ví dụ về đồ họa máy tính

*(*[*https://www.vietngapc.vn/upload\_images/case%20stream/Case%20Designer/designermonitor.jpg*](https://www.vietngapc.vn/upload_images/case%20stream/Case%20Designer/designermonitor.jpg)*)*

Thuật toán sắp xếp trong máy tính có tầm quan trọng rất lớn và được sử dụng rộng rãi trong các phần mềm phổ biến hiện nay. Để thuận tiện và giảm thiểu thời gian thao tác mà đặc biệt là để tìm kiếm dữ liệu dễ dàng và nhanh chóng. Do vậy, thao tác sắp xếp dữ liệu là một trong những thao tác cần thiết và thường gặp trong quá trình lưu trữ, quản lý dữ liệu.

## 1.1. Dev C++

Bloodshed Dev-C ++ (https://www.bloodshed.net/devcpp.html) là môi trường phát triển tích hợp (IDE) đầy đủ tính năng cho ngôn ngữ lập trình C/C ++ sử dụng Mingw của GCC (Bộ sưu tập trình biên dịch GNU) làm trình biên dịch.

Dev-C ++ cũng có thể kết hợp với Cygwin hoặc bất kỳ trình biên dịch dựa trên GCC nào khác. Các tính năng của Dev-C++:

* Hỗ trợ trình biên dịch dựa trên GCC.
* Gỡ lỗi tích hợp (sử dụng GDB- General DeBug).
* Quản lý dự án.
* Trình chỉnh sửa cú pháp.
* Trình duyệt lớp.
* Hoàn thành mã.
* Danh sách chức năng.
* Hồ sơ hỗ trợ.
* Nhanh chóng tạo Windows, console, thư viện tĩnh và DLL13.
* Hỗ trợ các mẫu để tạo các loại dự án của riêng bạn.
* Tạo Makefile.
* Chỉnh sửa và biên dịch các tệp Tài nguyên.
* Quản lý công cụ.
* Hỗ trợ in.
* Tìm và thay thế mã lệnh.
* Hỗ trợ CVS.

## 1.2. Thư viện Graphics.h

Vì sử dụng Dev C++ làm trình biên dịch cho việc cài đặt thuật toán nên không thể

thực hiện trên môi trường Windows. Vì vậy, một môi trường giả lập graphic của Borland C được Michael tạo ra thư viên có tên là Graphics.h. để có thể làm được điều đó. Micheal đã thay đổi BGI library (thư viện BGI) thành thư viện có tên WinBGIm để có thể sử dụng tốt trên windows. Và bây giờ bạn đã có thể sử dụng tốt các hàm đặc biệt của borland bằng Dev C++ (https://github.com/SagarGaniga/Graphics-Library)

## Thuật toán sắp xếp.

Sắp xếp là quá trình bố trí lại các phần tử của một tập đối tượng nào đó theo một thứ tự nhất định. Chẳng hạn như thứ tự tăng dần (hay giảm dần) đối với một dãy số, thứ tự từ điển đối với các từ v.v… Yêu cầu về sắp xếp thường xuyên xuất hiện trong các ứng dụng Tin học với các mục đích khác nhau: sắp xếp dữ liệu trong máy tính để tìm kiếm cho thuận lợi, sắp xếp các kết quả xử lý để in trên bảng biểu...

Dữ liệu có thể xuất hiện dưới nhiều dạng khác nhau, nhưng ở đây ta quy ước. Một tập các đối tượng cần sắp xếp là tập các bản ghi (records), mỗi bản ghi bao gồm một số trường (fields) khác nhau. Nhưng không phải toàn bộ các trường dữ liệu trong bản ghi đều được xem xét đến trong quá trình sắp xếp mà chỉ một trường nào đó (hay một vài trường hợp nào đó) được chú ý tới thôi. Trường như vậy ta gọi là **khóa (key)**. Sắp xếp sẽ được tiến hành dựa vào giá trị khóa này.

### Một số thuật toán sắp xếp.

Bubble-Sort: Thuật toán sắp xếp nổi bọt.

Select-Sort: Thuật toán sắp xếp kiểu chọn.

Cả hai thuật toán trên đều có độ phức tạp là O(n2). Select-Sort quá trình hoán đổi

ít hơn nhưng tốn thời gian tìm phần tử bé nhất.

Insert-Sort: Thuật toán sắp xếp kiểu chèn.

Do tận dụng được các dãy con đã được sắp xếp nên trong trường hợp tốt nhất độ phức tạp là O(n). Tuy nhiên trường hợp xấu nhất vẫn là O(n2).

Cả 3 thuật toán trên đều dựa vào 2 tương tác chính là so sánh và hoán đổi

Quick-Sort: Thuật toán sắp xếp kiểu phân đoạn. Là thuật toán sắp xếp tốt nhất hiện nay.

## 1.4. Thuật toán sắp xếp Quick-Sort

### 1.4.1. Sơ lược về thuật toán

Quick-Sort thuật toán được đề xuất bởi C.A.R Hoare – là một phương pháp sắp xếp tốt nhất, nghĩa là dù dãy có thứ tự sắp xếp như thế nào, Quicksort cũng có thể sắp xếp được và chưa có một thuật toán sắp xếp tổng quát nào nhanh hơn Quick-Sort về mặt tốc độ trung bình. Hoare đã mạnh dạn lấy chữ “Quick” để đặt tên cho thuật toán.

### 1.4.2. Ý tưởng chính của thuật toán

Dựa trên kỹ thuật chia để trị. Sắp xếp dãy khóa k[1…n] thì có thể coi là sắp xếp đoạn từ chỉ số 1 tới chỉ số n trong dãy khóa đó. Để sắp xếp một đoạn trong dãy khóa, nếu đoạn đó có ít hơn 2 khóa thì không cần phải làm gì cả, còn nếu đoạn đó có ít nhất 2 khóa, ta chọn một khóa ngẫu nhiên nào đó của đoạn làm “chốt” (Pivot). Mọi khóa nhỏ hơn khóa chốt được xếp vào vị trí đứng trước chốt, mọi khóa lớn hơn khóa chốt được xếp vào vị trí đứng sau chốt. Sau phép hoán chuyển như vậy thì đoạn đang xét được chia làm hai đoạn khác rỗng mà mọi khóa trong đoạn đầu đều ≤ chốt và mọi khóa trong đoạn sau đều ≥ chốt. Hay nói cách khác: Mỗi khóa trong đoạn đầu đều ≤ mọi khóa trong đoạn sau. Và vấn đề trở thành sắp xếp hai đoạn mới tạo ra (có độ dài ngắn hơn đoạn ban đầu) bằng phương pháp tương tự.

### 1.4.3. Ví dụ minh họa thuật toán

Giả sử ta cần sắp xếp mảng M có 10 phần tử sau (N = 10):

M: 45 55 25 20 15 5 25 30 10 3

Ban đầu: First = 1 Last = 10 X = M[(1 + 10)/2] = M[5] = 15

First Last

X = 15

15

M: 45 55 25 20 5 25 30 10 3

Phân hoạch:

X = 15

I J

15

M: 45 55 25 20 5 25 30 10 3

X = 15

I J

15

M: 3 55 25 20 5 25 30 10 45

X = 15

I J

15

M: 3 10 25 20 5 25 30 55 45

X = 15

I J

15

M: 3 10 5 20 25 25 30 55 45

X = 15

First I Last

15

M: 3 10 5 20 25 25 30 55 45

J

Phân hoạch các phần tử trong dãy con từ First -> J:

First = 1 Last = J = 4 X = M[(1 + 4)/2] = M[2] = 10

X = 10

First Last

10

M: 3 5 15 20 25 25 30 55 45

Phân hoạch:

X = 10

I J

10

M: 3 5 15 20 25 25 30 55 45

J

X = 10

10

M: 3 5 15 20 25 25 30 55 45

I

X = 10

J

10

M: 3 5 15 20 25 25 30 55 45

I

Phân hoạch các phần tử trong dãy con từ First ->J:

First = 1 Last = J = 2 X = M[(1 + 2)/2] = M[1] = 3

First Last

3

M: 5 10 15 20 25 25 30 55 45

X = 3

Phân hoạch:

I J

3

M: 5 10 15 20 25 25 30 55 45

X = 3

I=J

3

M: 5 10 15 20 25 25 30 55 45

X = 3

J I

3

M: 5 10 15 20 25 25 30 55 45

X = 3

First J I Last

M: 3 5 10 15 20 25 25 30 55 45

Phân hoạch các phần tử trong dãy con từ I -> Last:

First = I = 3 Last = 4 X = M[(3 + 4)/2] = M[3] = 10

First Last

10

M: 3 5 15 20 25 25 30 55 45

X = 10

Phân hoạch:

I J

10

M: 3 5 15 20 25 25 30 55 45

X = 10

I = J

10

M: 3 5 15 20 25 25 30 55 45

X = 10

X = 10

J I

10

M: 3 5 15 20 25 25 30 55 45

First J I Last

M: 3 5 10 15 20 25 25 30 55 45

Phân hoạch các phần tử trong dãy con từ I -> Last:

First = I = 5 Last = 10 X = M[(5 + 10)/2] = M[7] = 25

X = 25

First Last

25

M: 3 5 10 15 20 25 30 55 45

X = 25

I J

25

M: 3 5 10 15 20 25 30 55 45

I J

25

M: 3 5 10 15 20 25 30 55 45

X = 25

First J I Last

25

M: 3 5 10 15 20 25 30 55 45

X = 25

Phân hoạch các phần tử trong dãy con từ First -> J

First = 5 Last = J = 6 X = M[(5 + 6)/2] = M[5] = 20

First Last

20

M: 3 5 10 15 25 25 30 55 45

X = 20

Phân hoạch:

I J

20

M: 3 5 10 15 25 25 30 55 45

X = 20

I = J

20

M: 3 5 10 15 25 25 30 55 45

X = 20

J I

20

M: 3 5 10 15 25 25 30 55 45

X = 20

First J I Last

20

M: 3 5 10 15 25 25 30 55 45

Phân hoạch các phần tử trong dãy con từ I -> Last:

First = I = 7 Last = 10 X = M[(7 + 10)/2] = M[8] = 30

First Last

30

M: 3 5 10 15 20 25 25 55 45

X = 30

Phân hoạch:

I J

30

M: 3 5 10 15 20 25 25 55 45

X = 30

I = J

30

M: 3 5 10 15 20 25 25 55 45

X = 30

J I

30

M: 3 5 10 15 20 25 25 55 45

X = 30

First = J I Last

30

M: 3 5 10 15 20 25 25 55 45

X = 30

Phân hoạch các phần tử trong dãy con từ I -> Last:

First = I = 9 Last = 10 X = M[(9 + 10)/2] = M[9] = 55

First Last

55

M: 3 5 10 15 20 25 25 30 45

X = 55

Phân hoạch:

I J

55

M: 3 5 10 15 20 25 25 30 45

X = 55

J I

55

M: 3 5 10 15 20 25 25 30 45

X = 55

M: 3 5 10 15 20 25 25 30 45 55

Toàn bộ quá trình phân hoạch kết thúc, dãy M trở thành:

M: 3 5 10 15 20 25 25 30 45 55

### 1.4.4. Độ phức tạp giải thuật

Gọi T(n) là thời gian thực hiện giải thuật.

Trong trường hợp tốt nhất: Mỗi lần phân chia dãy con tạo thành 2 dãy con có độ dài bằng nhau.

T(n) = nt0 + T( T( (Với t0: Là thời gian duyệt thuật toán)

= nt0 + 2 T(

= nt0 + 2 [(t0 + 2 T(] (T(n) = nt0 + 2 T( )

= nt0 + nt0 + 22 T(

= 2nt0 + 22 T(

= 2nt0 + 22 [( t0 + 2 T(]

= 2nt0 + nt0 + 23 T(

= 3nt0 + 23 T(

= knt0 + 2k T( (Với k Z)

Thuật toán dừng khi:

= 1 => n = => k =

T(n) = (n nt0 + nT(1)

= O(n)

Trong trường hợp xấu nhất: Yêu cầu sắp xếp tăng dần mà dãy Input có thứ tự giảm dần và ngược lại.

T(n) = nt0 + T(1) + T(n – 1)

= nt0 + T(n – 1)

= nt0 + (n – 1) t0 + T(n – 2)

= nt0 + (n – 1) t0 + … + 2 t0 + T(1)

= t0 (n + (n – 1) + … + 2)

= t0 (n + 2)(n – 1)/2 = O(n2)

**Nhận xét:**

Thuật toán sắp xếp Quicksort tốt hơn 3 thuật toán sắp xếp cơ bản.

Về mặt toán học, người ta chứng minh được: Nếu sắp xếp trên mảng bằng cách so sánh và hoán đổi, thì độ phức tạp bé nhất là: T(n) = O(n)

Nhược điểm: Phụ thuộc vào cách chọn phần tử chốt (pivot). Không ổn định.

### 1.4.5. Lưu đồ giải thuật

Hình 1.2. Lưu đồ giải thuật Quick-Sort

# CHƯƠNG 2. PHƯƠNG PHÁP NGHIÊN CỨU

## 2.1. Cài đặt Dev C và thư viện graphics.h

Tải file cài đặt phần mềm Dev C++ theo đường dẫn trong mục 1.4. Sau đó mở file

vừa tải, và tiến hành cài đặt. Thư viện graphics.h được tiến hành cài đặt theo các bước:

**Bước 1:** Copy 6-ConsoleAppGraphics và ConsoleApp\_cpp\_graph

Paste C:\...\Dev-Cpp\Templates

**Bước 2:** Copy graphics và winbgim

Paste C:\... Dev-Cpp \MinGW64\x86\_64-w64-mingw32\include

**Bước 3:** Copy libbgi.a

Paste C:\...\Dev-Cpp\MinGW64\x86\_64-w64-mingw32\lib

**Bước 4:** Ở DevC++ => New Project => Console Graphics Application

**Bước 5:** Thay đổi Tools - Compiler Option: TDM – GCC 4.9.2 32 bit Release

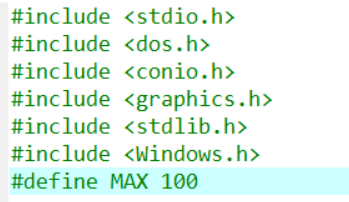
**Bước 6:** Sử dụng đoạn code mẫu bên dưới để test thư viện winbgim

|  |  |
| --- | --- |
| Capture | Capture2 |

Hình 2.1. Ví dụ minh hoạ thư viện winbgim

## 2.2. Cài đặt thuật toán tô màu

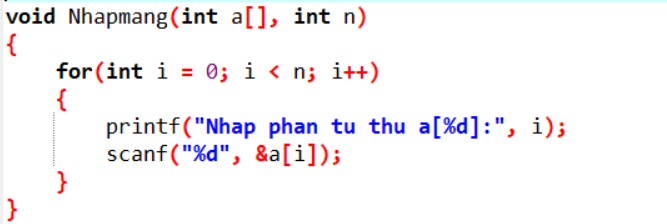
***Khai báo thư viện***



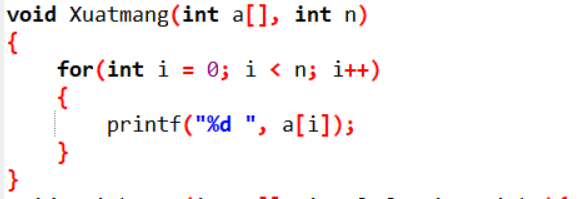
**Khai báo biến**



**Tạo mảng:** Tạo mảng gồm 12 phần tử.

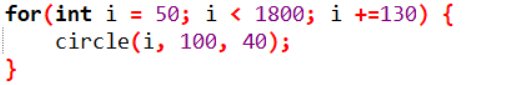


**Xuất mảng:**

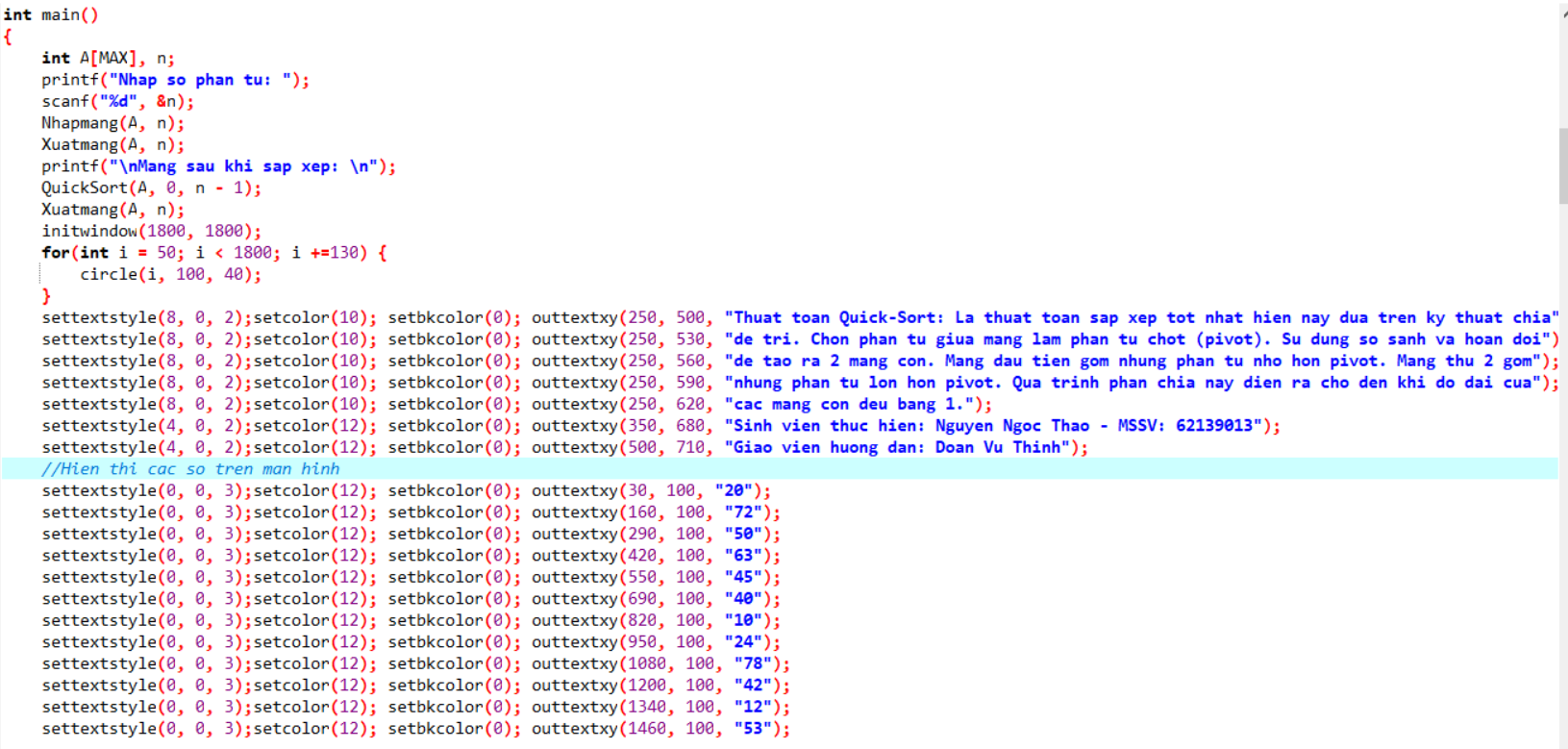


Kết thúc việc nhập mảng sẽ tiến hành vẽ hình tròn cho mỗi số

**Thuật toán vẽ hình tròn:**

****

***Chương trình chính***

******

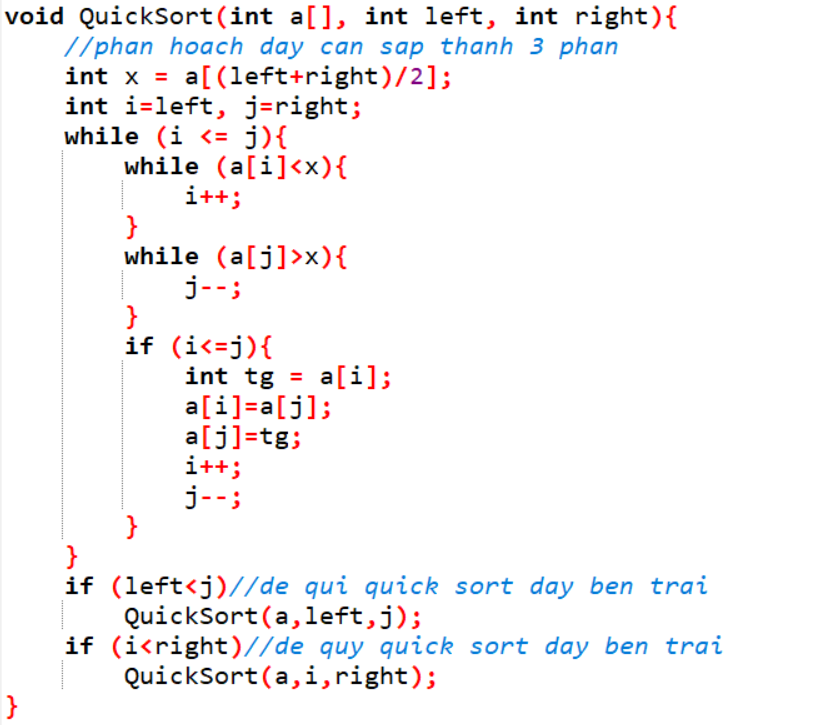
## 2.3. Cài đặt thuật toán Quick-Sort

***Khai báo thư viện***(như trên)

***Khai báo biến***

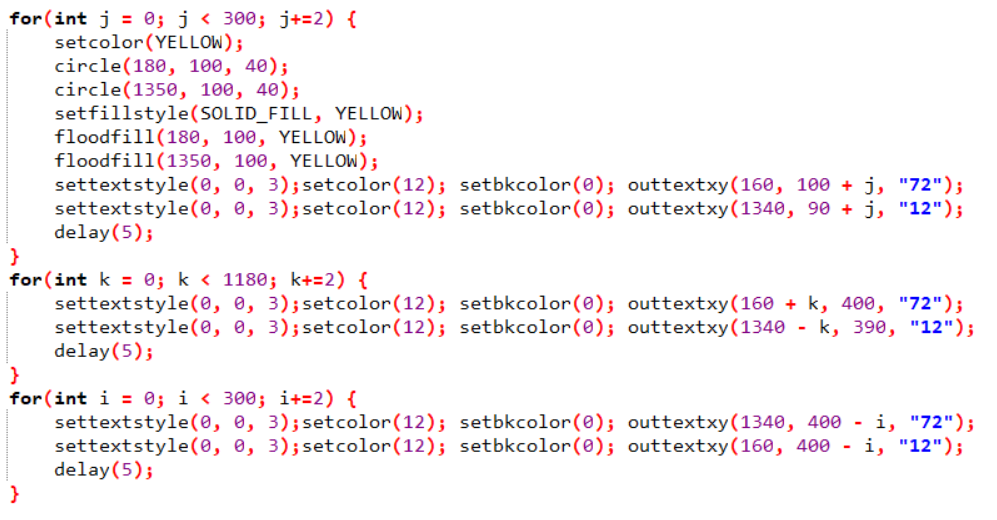
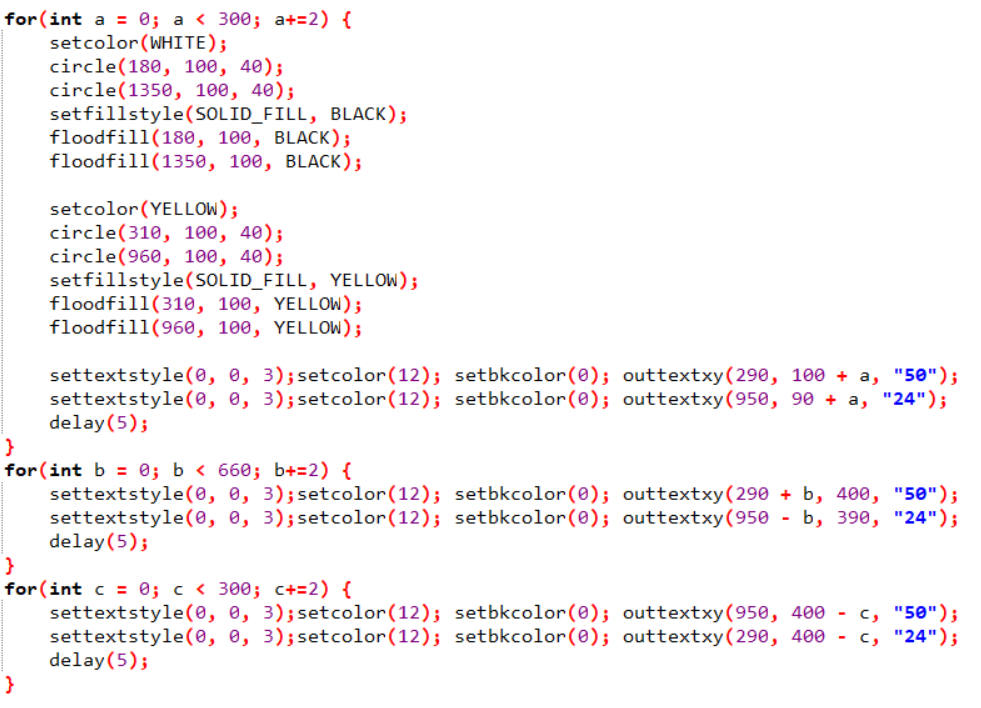


**Thực hiện sắp xếp**

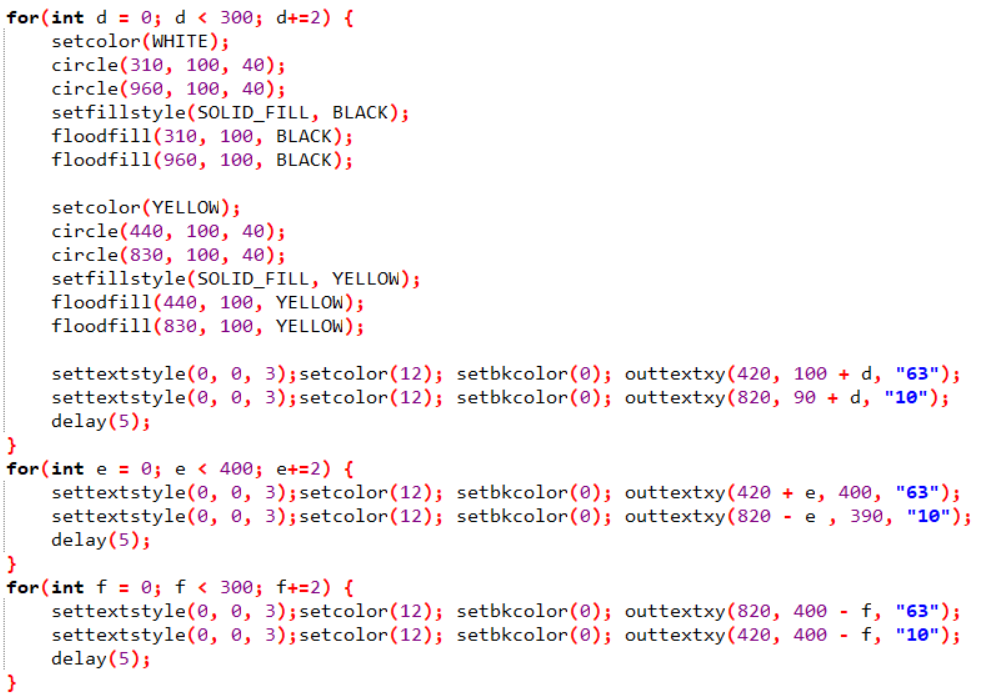
****

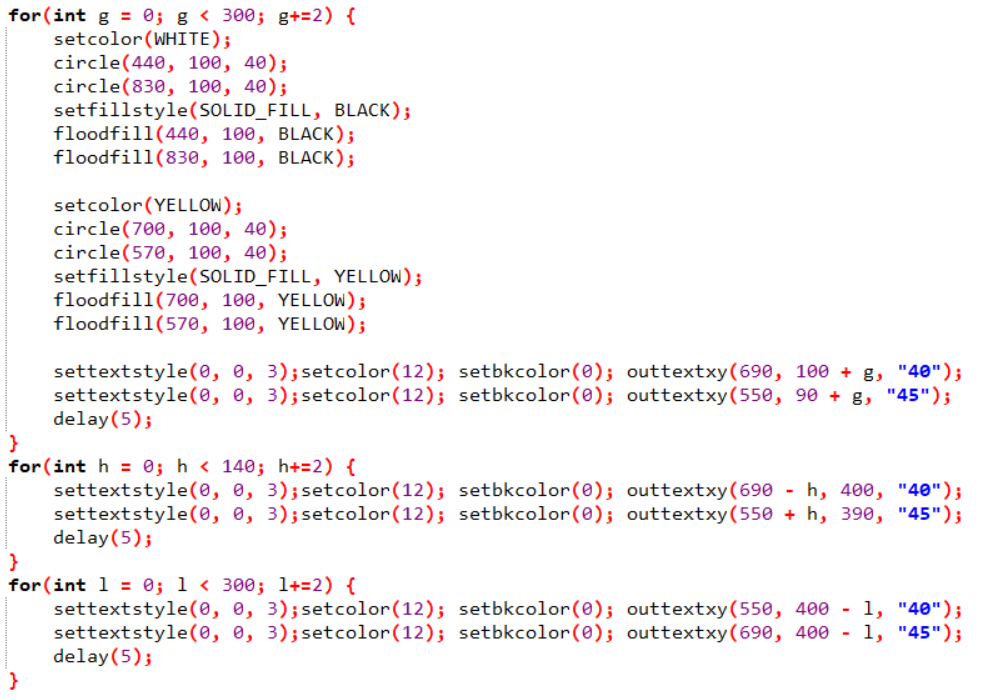
## 2.3. Thực hiện phép tịnh tiến

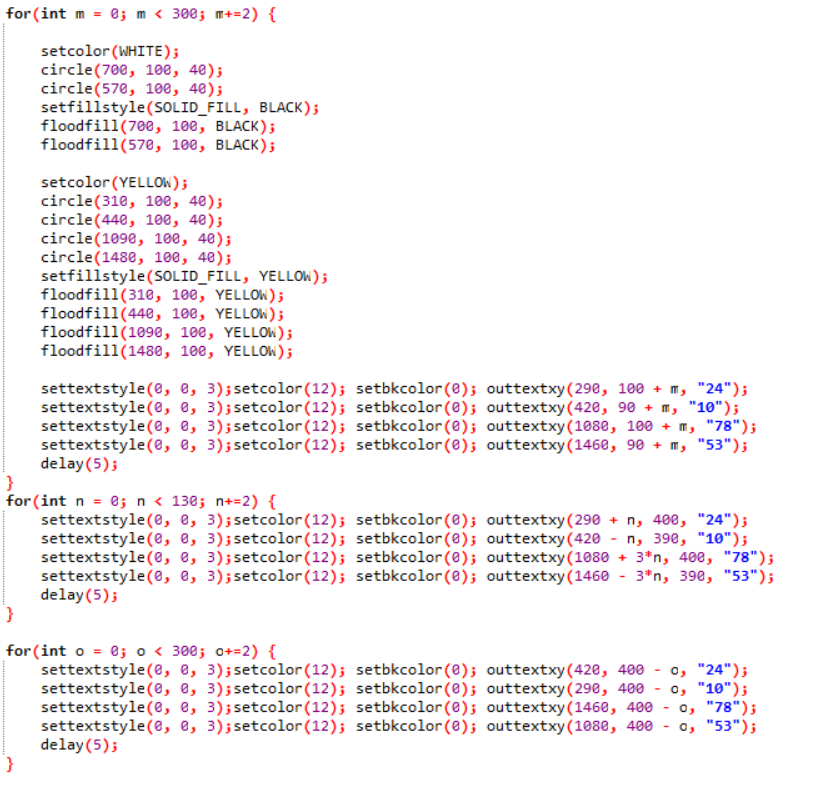
Đổi chỗ 2 số 72 và 12.

 Đổi chỗ 2 số 50 và 24.

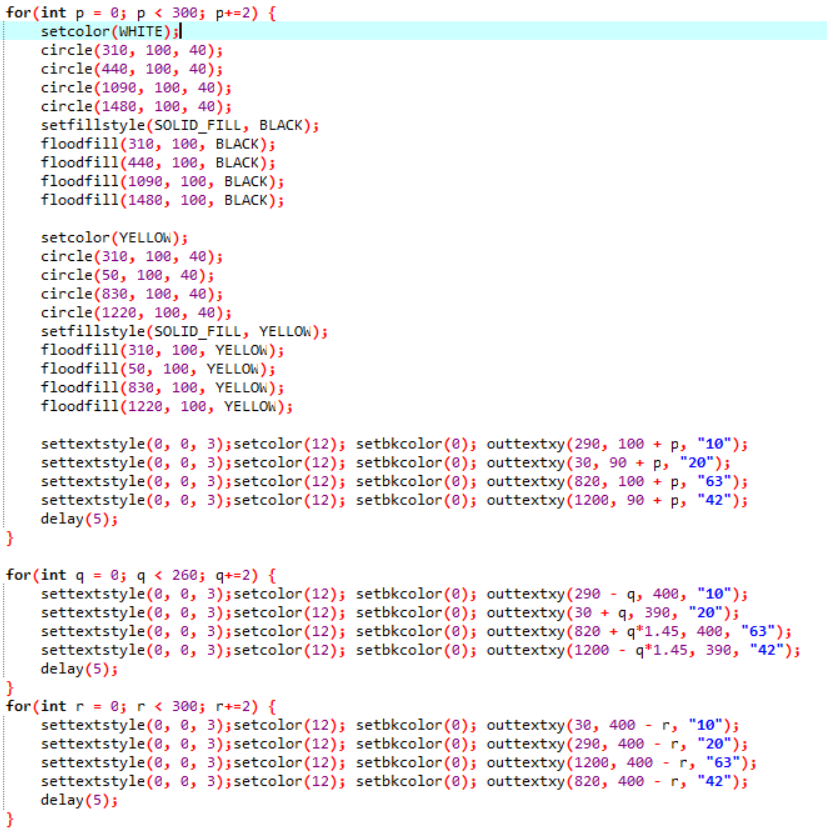
Đổi số 2 số 63 và 10.



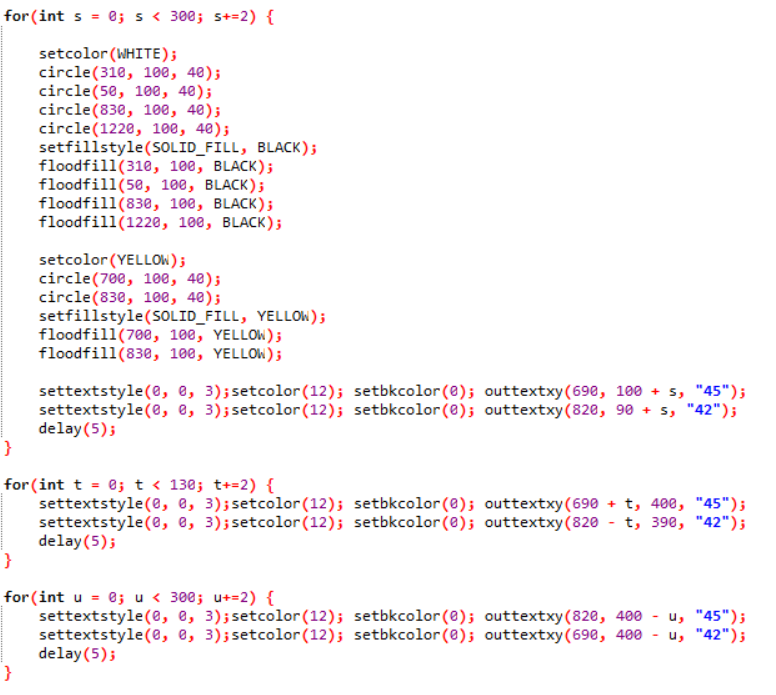
 Đổi chỗ 2 số 40 và 45.

 Đổi chỗ 2 số 24 với 10 và 78 với 53.

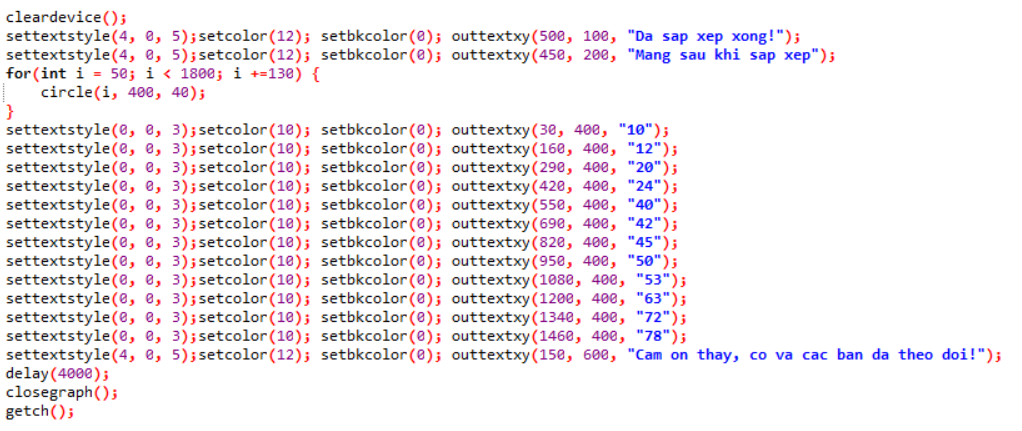
Đổi chỗ 2 số 10 với 20 và 63 với 42.



Đổi chỗ 2 số 45 với 42.



## 2.3. Hiển thị kết quả



# CHƯƠNG 3. KẾT QUẢ THỰC HIỆN

## 3.1. Mô phỏng thuật toán



 Sử dụng phép tịch tiến bằng cách tăng giá trị tung độ đến vị trí 400 pixel.

Sử dụng phép tịnh tiến bằng cách tăng hoành độ đến vị trí 1340 pixel đối với số 72 và giảm hoành độ đến vị trí 160 pixel đối với số 12.



Sử dụng phép tịnh tiến bằng cách giảm tung độ đến vị trí 100 pixel.



Như vậy, đã hoàn thành việc sắp xếp bằng cách hoán đổi vị trí 2 số 12 và 72. Các số còn lại tương tự.

## 3.1. Kết thúc mô phỏng



Hình 3.1. Kết quả sau khi sắp xếp

Sau khi sắp xếp xong thực hiện việc in ra màn hình mảng sau khi sắp xếp.

# THẢO LUẬN

Đề tài đã mô phỏng được thuật toán sắp xếp Quick-Sort với dữ liệu cho trước. Ngoài ra còn tìm hiểu thêm về độ phức tạp giải thuật của một số thuật toán sắp xếp khác. Thấy được nhược điểm của thuật toán sắp xếp Quick-Sort so với các thuật toán sắp xếp cơ bản khác là phụ thuộc vào cách chọn phần tử chốt và tính không ổn định của thuật toán.

Tuy nhiên, đề tài vẫn chưa cài đặt được thuật toán sắp xếp Quick-Sort với việc nhập dữ liệu từ bàn phím và đồng thời chưa khắc phục được hai nhược điểm lỗi dư thừa điểm ảnh và tô màu trong lúc di chuyển của các con số.

Trong tương lai, em sẽ tiếp tục tìm hiểu thêm và cài đặt thuật toán sắp xếp Quick-Sort nhập dữ liệu từ bàn phím và sẽ cố gắng tìm hiểu và cải tiến 2 nhược điểm trên trong quá trình lập trình.

Qua đợt thực tập cơ sở, em đã có kiến thức sử dụng thư viện đồ họa Graphics.h. Trong tương lai, em sẽ sử dụng các kiến thức này để lập trình trò chơi đơn giản bằng thư viện đồ họa Graphics.h.

# TÀI LIỆU THAM KHẢO

1. Nguyễn Quang Khánh, “Đồ họa máy tính”, 2015, NXB Khoa học Kỹ thuật
2. Vũ Hải Quân, “Đồ họa máy tính”, 2007, NXB Đại Học Quốc Gia
3. D. Hearn, M.P. Baker, “Computer Graphics, C version”, 1997, Prentice Hall
4. Đoàn Vũ Thịnh, “Bài giảng Kỹ thuật đồ họa”, 2019, Đại học Nha Trang
5. Lê Minh Hoàng, “Giải thuật & Lập trình”, 1999-2006, Đại học Sư phạm Hà Nội
6. Data Structure and Algorithmic Thinking with Python, 2016, Narasimha Karumanchi