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 TÌM KIẾM NHỊ PHÂN TRONG MẢNG**

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

**Sinh viên thực hiện: Nguyễn Văn Tín**

**Mã số sinh viên: 62134339**

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 TÌM KIẾM NHỊ PHÂN TRONG MẢNG**

Giảng viên hướng dẫn: ThS. Đoàn Vũ Thịnh

Sinh viên thực hiện: Nguyễn Văn Tín

Mã số sinh viên: 62134339

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 TÌM KIẾM NHỊ PHÂN TRONG MẢNG**

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

**Sinh viên được hướng dẫn:** Nguyễn Văn Tín

**MSSV:** 62134339

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Lần** | **Ngày** | **Nội dung** | **Nhận xét của GVHD** |
| 1 | 2/12/2022 | Nhận đề tài hướng dẫn và định hướng giải quyết vấn đề. Sinh viên trình bày kế hoạch thực hiện. |  |
| 2 | 6/12/2022 | 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. Đã làm được chương trình của thuật toán nhưng chưa được đồ họa graphics.h, chưa nhập dữ liệu đầu vào từ bàn phím và giới hạn các phần từ nhập vào. |  |
| 3 | 14/12/2022 | Sinh viên hoàn thiện các thuật toán đã đề ra với dữ liệu đầu vào được nhập từ bàn phím và đã giới hạn được số phần từ nhập vào từ bàn phím. Chưa làm được đồ họa graphics.h , làm được sắp xếp mảng tăng dần và tìm được vị trí của từng số trong mảng. |  |
| 4 | 21/12/2022 | Sinh viên nộp bản thảo của báo cáo thực tập lần thứ 1 và tiến hành chỉnh sửa. Đã làm được đồ họa graphics.h sắp xếp mảng và đã có được tên sinh viên và giáo viên hướng dẫn trên màn hình đồ họa. |  |
| 5 | 28/01/2022 | Sinh viên nộp bản thảo lần 2 và có minh họa với thư viện của chuột nhưng chưa thể kết nối với phần trước. |  |
| 6 | 5/1/2023 | Sinh viên nộp bản thảo lần cuối sau khi đã chỉnh sửa các yêu cầu như đã đề ra. |  |

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

Sinh viên thực hiện tốt các yêu cầu của GVHD, trong quá trình thực hiện đề tài có sự liên hệ chặt chẽ với GV. Theo lịch hẹn Sinh viên đều có mặt để trình bày ý tưởng của các nội dung lần trước. Trong quá trình hoàn tất báo cáo đều nỗ lực không ngừng mặc dù đang cao điểm của đợt thi học kỳ nhưng SV vẫn dành thời gian không ít cho TTCS.

Về nội dung báo cáo đã thỏa mãn các yêu cầu của đề tài như trong đề cương. Về kết quả chương trình đã minh họa được thuật toán. Về các yêu cầu cao hơn như sử dụng thư viện của chuột hay minh họa các trường hợp nhược điểm của thuật toán thì chưa thực hiện thành công.

Về hình thức của báo cáo và sản phẩm, báo cáo trình bày rõ ràng các mục tiêu, phương pháp, kết quả và thảo luận cho sản phẩm. Còn về sản phẩm như đã trình bày có phần hạn chế.

Điểm hình thức: Điểm nội dung: **Điểm tổng kết:** Đồng ý cho sinh viên: Được bảo vệ: 🗹 Không được bảo vệ: 

Khánh Hòa, ngày 6 tháng 01 năm 2023

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](#_Toc123764809)

[MỤC LỤC ii](#_Toc123764810)

[TÓM TẮT iii](#_Toc123764811)

[GIỚI THIỆU 1](#_Toc123764812)

[*1.1. Thuật toán tìm kiếm nhị phân (Binary Search):* 2](#_Toc123764813)

[*1.1.1. Khái niệm và ý tưởng:* 2](#_Toc123764814)

[*1.1.2. Khái quát thuật toán:* 3](#_Toc123764815)

[*1.1.3.* *Cách thức hoạt động của thuật toán:* 3](#_Toc123764816)

[Cách Binary Search làm việc: 3](#_Toc123764817)

[*1.2. Hiển thị số lên màn hình:* 6](#_Toc123764818)

[*1.3. Dev C++* 7](#_Toc123764819)

[*1.4. Thư viện Graphics.h* 7](#_Toc123764820)

[CHƯƠNG 2. PHƯƠNG PHÁP NGHIÊN CỨU 8](#_Toc123764821)

[*2.1. Cài đặt DevC và thư viện graphics.h* 8](#_Toc123764822)

[*2.2. Cài đặt thuật toán tìm kiếm nhị phân (Binary Search):* 8](#_Toc123764823)

[*2.3. Cài đặt thuật toán bằng thư viện graphics.h* 10](#_Toc123764824)

[CHƯƠNG 3. KẾT QUẢ THỰC HIỆN 14](#_Toc123764825)

[*3.1. Hình ảnh thuật toán tìm kiếm nhị phân trong mảng* 14](#_Toc123764826)

[*3.2. Hình ảnh mảng chưa sắp xếp trong thư viện graphics.h* 15](#_Toc123764827)

[THẢO LUẬN 16](#_Toc123764828)

[TÀI LIỆU THAM KHẢO 17](#_Toc123764829)

# **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 tìm kiếm nhị phân là một trong những thuật toán tìm kiếm quan trọng nhất của tin học. Thuật toán này còn được gọi là thuật toán chặt nhị phân hay thuật toán chia đôi được áp dụng rất nhiều trong giải toán, nó làm giảm được nhiều thời gian tìm kiếm, giúp chương trình chạy nhanh hơn.

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 DevC/C++ có kết hợp với thư viện graphics.h.

Sản phẩm đã minh họa được từng bước tìm kiếm nhị phân trong mảng. 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 nhập các số ngẫu nhiên từ bàn phím và tự động sắp xếp và tìm kiếm. 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:

# 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, chuẩn 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

Thuật toán tìm kiếm nhị phân là một trong những thuật toán tìm kiếm quan trọng nhất của tin học. Thuật toán này còn được gọi là thuật toán chặt nhị phân hay thuật toán chia đôi được áp dụng rất nhiều trong giải toán, nó làm giảm được nhiều thời gian tìm kiếm, giúp chương trình chạy nhanh hơn.

## *1.1. Thuật toán tìm kiếm nhị phân (Binary Search):*

## *1.1.1. Khái niệm và ý tưởng:*

**Khái niệm:**

Binany Search (Tìm kiếm nhị phân) là một [giải thuật](https://quantrimang.com/cong-nghe/cau-truc-du-lieu-va-giai-thuat) tìm kiếm nhanh với độ phức tạp thời gian chạy là Ο(log n). Giải thuật tìm kiếm nhị phân làm việc dựa trên nguyên tắc chia để trị (Divide and Conquer). Để giải thuật này có thể làm việc một cách chính xác thì tập dữ liệu nên ở trong dạng đã được sắp xếp.

Binary Search tìm kiếm một phần tử cụ thể bằng cách so sánh phần tử tại vị trí giữa nhất của tập dữ liệu. Nếu tìm thấy kết nối thì chỉ mục của phần tử được trả về. Nếu phần tử cần tìm là lớn hơn giá trị phần tử giữa thì phần tử cần tìm được tìm trong mảng con nằm ở bên phải phần tử giữa; nếu không thì sẽ tìm ở trong mảng con nằm ở bên trái phần tử giữa. Tiến trình sẽ tiếp tục như vậy trên mảng con cho tới khi tìm hết mọi phần tử trên mảng con này.

Người ta đã chứng minh được độ phức tạp của thuật toán tìm kiếm nhị phân trong trường hợp tốt nhất là O(1), trong trường hợp xấu nhất là O(log2n) và trung bình cũng là O(log2n).

**Ý tưởng:**

Tìm kiếm nhị phân là một thuật toán tìm kiếm giá trị xác định trên dãy đã sắp xếp.

Thuật toán ban đầu yêu cầu mảng đã được sắp xếp.

Thuật toán tiến hành so sánh giá trị cần tìm với phần tử đứng giữa mảng. Lúc này mảng sẽ được chia ra làm 2 phần bên trái và bên phải phần tử đó.

Nếu chúng không bằng nhau, phần nửa mảng không chứa giá trị cần tìm sẽ bị bỏ qua và tiếp tục tìm kiếm trên nửa còn lại, một lần nữa lấy phần tử ở giữa và so sánh với giá trị cần tìm, cứ thế lặp lại cho đến khi tìm thấy giá trị đó.

Nếu tìm kiếm kết thúc khi nửa còn lại trống thì ta kết luận giá trị không có trong mảng.

Có vẻ vẫn còn hơi khó hình dung, để hiểu rõ hơn về thuật toán, chúng ta sẽ tới các bước thực hiện sau đây:

- Đặt 2 biến đánh dấu vị trí đầu và cuối của phạm vi trong mảng cần xét.

- Chia mảng thành 2 phần bằng nhau và so sánh giá trị cần tìm với giá trị nằm giữa 2 phần đó. Nếu giá trị cần tìm lớn hơn, ta xét nửa sau của mảng. Nếu nhỏ hơn, thực hiện xét nửa trước của mảng. Ta cập nhật lại 2 biến đánh dấu phạm vi mảng cần xét. Nếu bằng, ta xác định được vị trí của phần tử và kết thúc thuật toán.

## *1.1.2. Khái quát thuật toán:*

- Thuật toán tìm kiếm nhị phân hoạt động trên các mảng đã được sắp xếp. Thuật toán bắt đầu bằng việc so sánh một phần tử đứng chính giữa mảng với giá trị cần tìm. Nếu bằng nhau, vị trí của nó trong mảng sẽ được trả về. Nếu giá trị cần tìm nhỏ hơn phần tử này, quá trình tìm kiếm tiếp tục ở nửa nhỏ hơn của mảng.

- Nếu giá trị cần tìm lớn hơn phần tử ở giữa, quá trình tìm kiếm tiếp tục ở nửa lớn hơn của mảng. Bằng cách này, ở mỗi phép lặp thuật toán có thể loại bỏ nửa mảng mà giá trị cần tìm chắc chắn không xuất hiện

## *1.1.3. Cách thức hoạt động của thuật toán:*

## Cách Binary Search làm việc:

Để Binary Search làm việc thì mảng phải cần được sắp xếp. Để tiện cho việc theo dõi, mình sẽ cung cấp thêm các hình minh họa tương ứng với mỗi bước.

Giả sử chúng ta cần tìm vị trí của giá trị 31 trong một mảng bao gồm các giá trị như hình dưới đây bởi sử dụng Binary Search:



Đầu tiên, chúng ta chia mảng thành hai nửa theo phép toán sau:

chỉ-mục-giữa = ban-đầu + (cuối + ban-đầu)/ 2

Với ví dụ trên là 0 + (9 – 0)/ 2 = 4 (giá trị là 4.5). Do đó 4 là chỉ mục giữa của mảng.

Bây giờ chúng ta so sánh giá trị phần tử giữa với phần tử cần tìm. Giá trị phần tử giữa là 27 và phần tử cần tìm là 31, do đó là không kết nối. Bởi vì giá trị cần tìm là lớn hơn nên phần tử cần tìm sẽ nằm ở mảng con bên phải phần tử giữa.



Chúng ta thay đổi giá trị ban-đầu thành chỉ-mục-giữa + 1 và lại tiếp tục tìm kiếm giá trị chỉ-mục-giữa.

ban-đầu = chỉ-mục-giữa + 1chỉ-mục-giữa = ban-đầu + (cuối + ban-đầu)/ 2

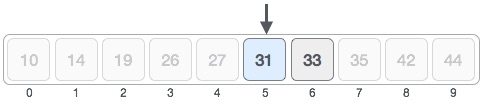
Bây giờ chỉ mục giữa của chúng ta là 7. Chúng ta so sánh giá trị tại chỉ mục này với giá trị cần tìm.



Giá trị tại chỉ mục 7 là không kết nối, và ngoài ra giá trị cần tìm là nhỏ hơn giá trị tại chỉ mục 7 do đó chúng ta cần tìm trong mảng con bên trái của chỉ mục giữa này.



Tiếp tục tìm chỉ-mục-giữa lần nữa. Lần này nó có giá trị là 5.



So sánh giá trị tại chỉ mục 5 với giá trị cần tìm và thấy rằng nó kết nối.

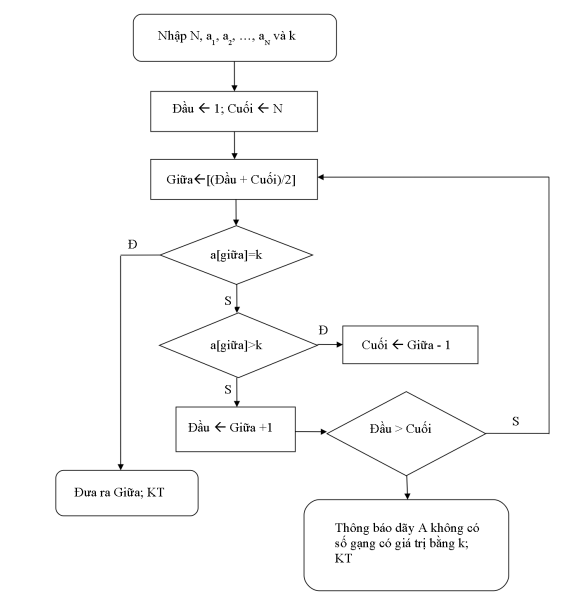


Do đó chúng ta kết luận rằng giá trị cần tìm 31 được lưu giữ tại vị trí chỉ mục 5.

Binary Search chia đôi lượng phần tử cần tìm và do đó giảm số lượng phép so sánh cần thực hiện nên giải thuật tìm kiếm này được thực hiện khá nhanh.

**Lưu đồ thuật toán:**

Bước 1: Nhập N, các số hạng a1, a2,...,aN và khóa k;  
Bước 2: Dau := 1, Cuoi := N;  
Bước 3: Giua := [(Dau + Cuoi)/2];  
Bước 4: Nếu aGiua = k thì thông báo chỉ số Giua, rồi kết thúc;  
Bước 5: Nếu aGiua > k thì đặt Cuoi = Giua - 1, rồi chuyển đến bước 7;  
Bước 6: Dau := Giua + 1;  
Bước 7: Nếu Dau > Cuoi thì thông báo dãy A không có số hạng có giá trị bằng k, rồi kết thúc;  
Bước 8: Quay lại bước 3.

****

***Bài toán ví dụ:***

Một bài toán tương đối quen thuộc khi học về thuật toán: Cho 1 mảng đã được sắp xếp, tìm vị trí của số k trong mảng.

ex: Mảng 1, 2, 4, 7, 9

Tìm vị trí của số 4 => vị trí 3

Input:

- Dòng đầu chứa 2 số tự nhiên n (số phần tử của mảng) và k (vị trí cần tìm)

- Dòng 2 chứa n số tự nhiên tăng dần

Output: số thứ k trong dãy

Hướng giải: Chúng ta có thể duyệt từ đầu mảng đến cuối mảng nếu phần tử nào bằng giá trị thì trả về phần tử đó. Nhưng cách này có độ phức tạp là O(n) => chỉ nên sử dụng khi số lượng nhỏ

## *1.2. Hiển thị số lên màn hình:*

Để viết chữ trong chế độ đồ hoạ thì trước tiên ta phải chọn font chữ, cỡ chữ, hướng in, căn chỉnh chiều dọc hay ngang đối với điểm in.

– Hàm void settextstyle(int phông, int hướng, int cỡ): hàm thiết đặt font chữ, hướng in, cỡ chữ.

– Hàm void settextstyle(int phông, int hướng, int cỡ): hàm thiết đặt font chữ, hướng in, cỡ chữ.  
 + Tham số phông quy định font chữ, các giá trị sau:  
 + Tham số hướng quy định hướng in, nhận các giá trị  
HORIZ\_DIR (0) : in ngang (giá trị ngầm định).  
VERT\_DIR (1) : in dọc.  
 + Tham số cỡ quy định cỡ chữ, có giá trị từ 1 đến 10 (to nhất).

– Hàm void settextjustify(int ngang, int dọc) : quy định nơi hiển thị văn bản của outtextxy theo quan hệ với toạ độ (x,y), của outtext theo quan hệ với vị trí hiện tại của con trỏ (giả sử (x,y)).  
 + Tham số ngang có giá trị  
 + Tham số dọc có các giá trị  
 + Hàm outtextxy(int x, int y, char \*s) : in xâu s theo vị trí (x,y).

**Ví dụ:** settextstyle(0, 0, 2);setcolor(12); setbkcolor(0); outtextxy(400, 600, "Sinh vien thuc hien: Nguyen Van Tin - MSSV: 62134339");

settextstyle(0, 0, 2);setcolor(12); setbkcolor(0); outtextxy(500, 630, "Giao vien huong dan: Doan Vu Thinh");

## *1.3. 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.

## *1.4. Thư viện Graphics.h*

Vì sử dụng DevC++ 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 DevC++ (https://github.com/SagarGaniga/Graphics-Library)

# CHƯƠNG 2. PHƯƠNG PHÁP NGHIÊN CỨU

## *2.1. Cài đặt DevC và thư viện graphics.h*

Tải file cài đặt phần mềm DevC++ 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 - Complier Option: TDM – GCC 4.9.2 32 bit Release

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

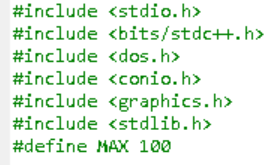
|  |  |
| --- | --- |
| Capture | Capture2 |

Hình 2.1. Ví dụ minh hoạ thư viện winbgim

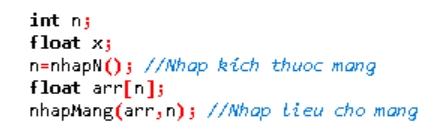
## 

## *2.2. Cài đặt thuật toán tìm kiếm nhị phân (Binary Search):*

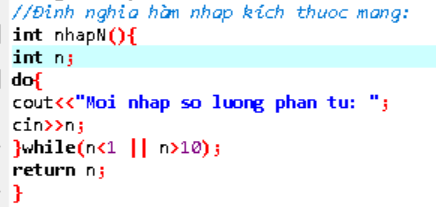
***Khai báo thư viện***



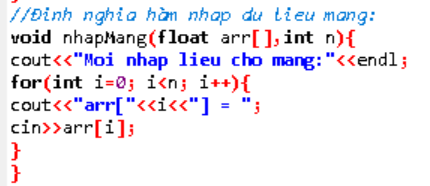
**Khai báo biến**



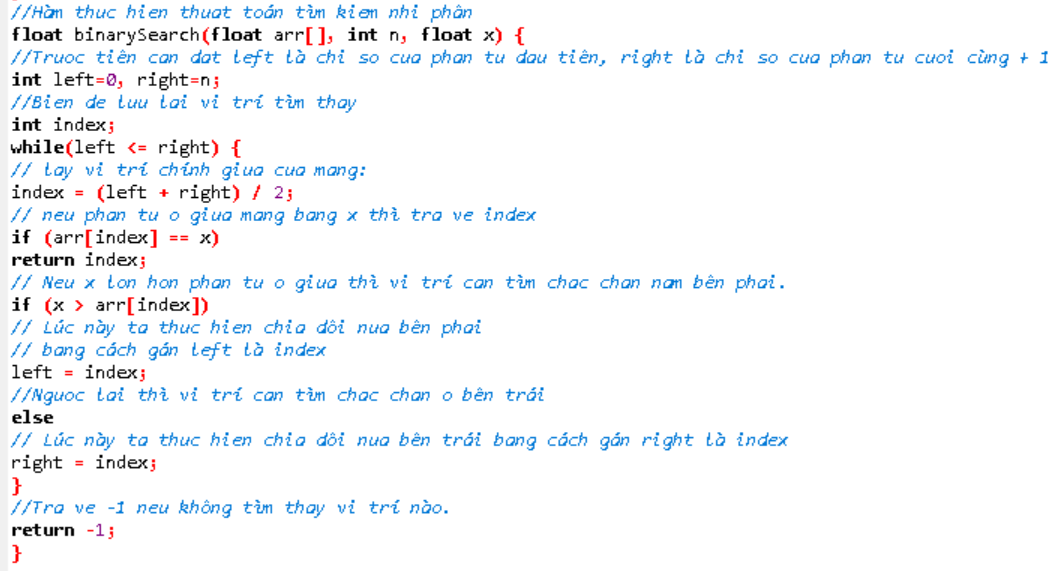
***Ðịnh nghĩa hàm nhập kích thước mảng***

******

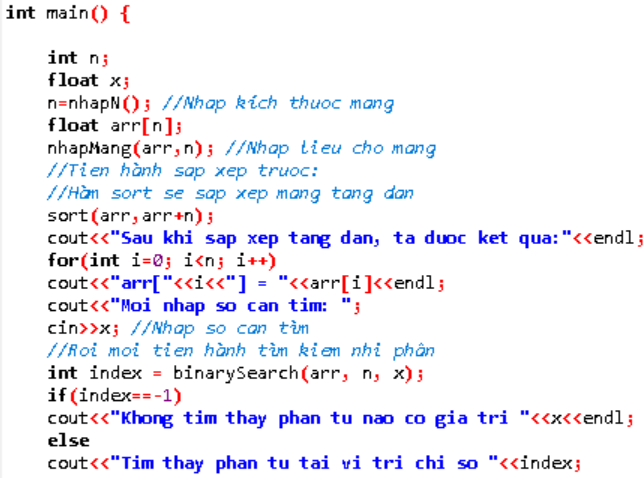
***Định nghĩa hàm nhập dữ liệu mảng***

******

***Hàm thực hiện thuật toán tìm kiếm nhị phân***

******

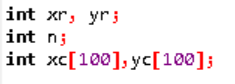
***Chương trình chính***



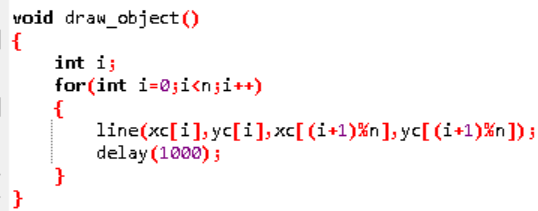
## *2.3. Cài đặt thuật toán bằng thư viện graphics.h*

***Khai báo thư viện***(như trên)

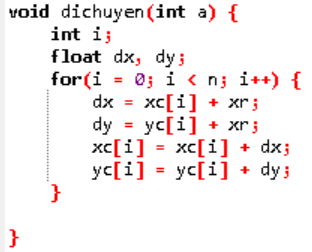
***Khai báo biến***



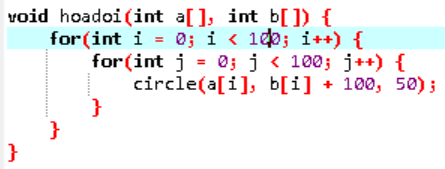
**Hàm vẽ đối tượng**

****

**Hàm di chuyển**

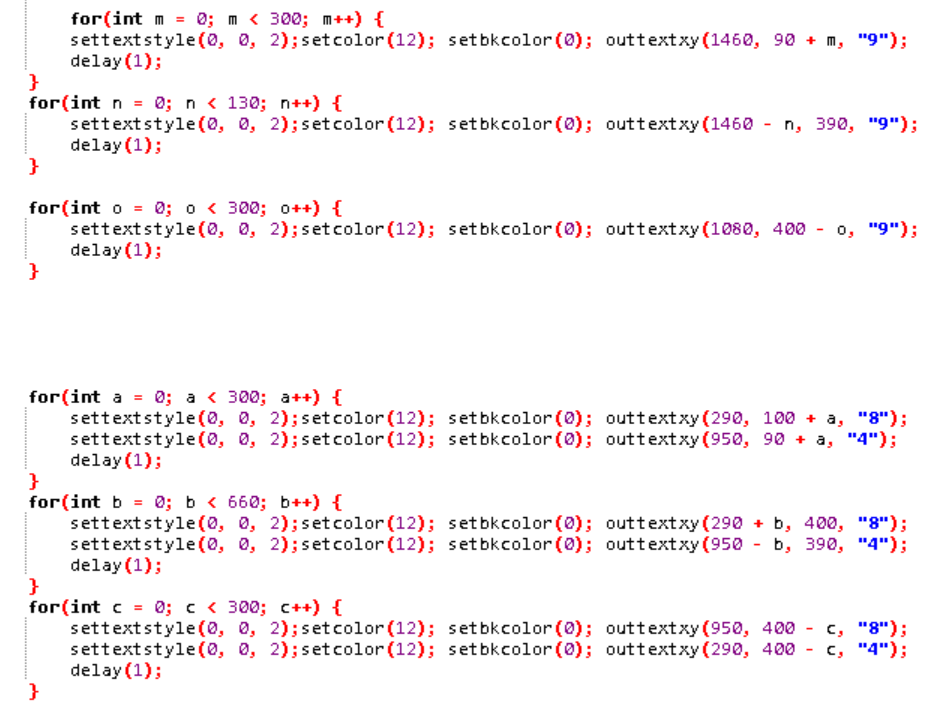
****

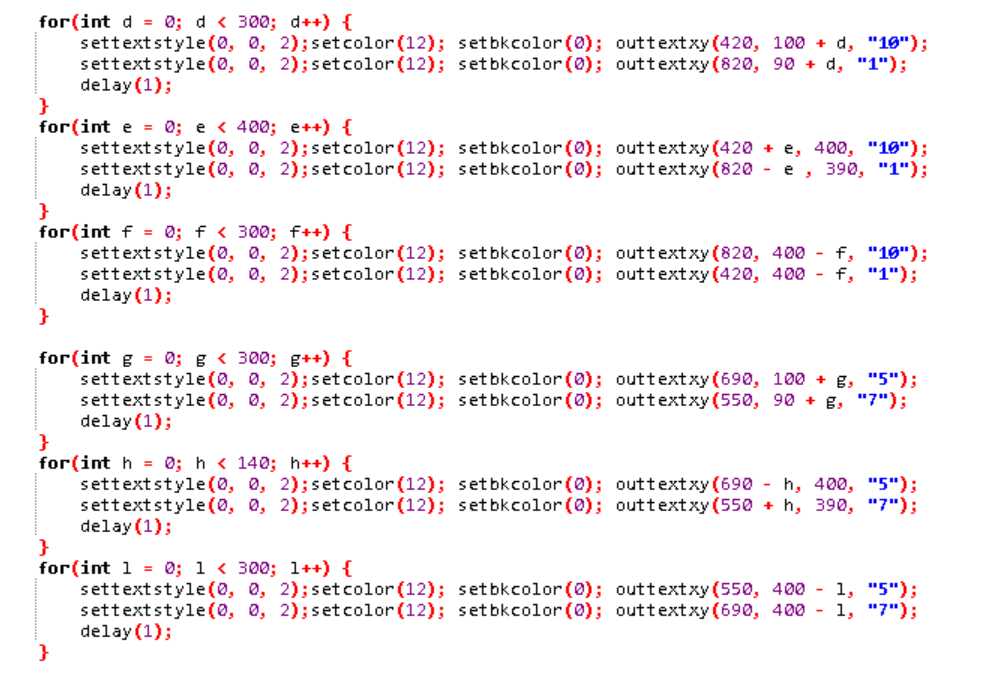
**Hàm hoan doi**

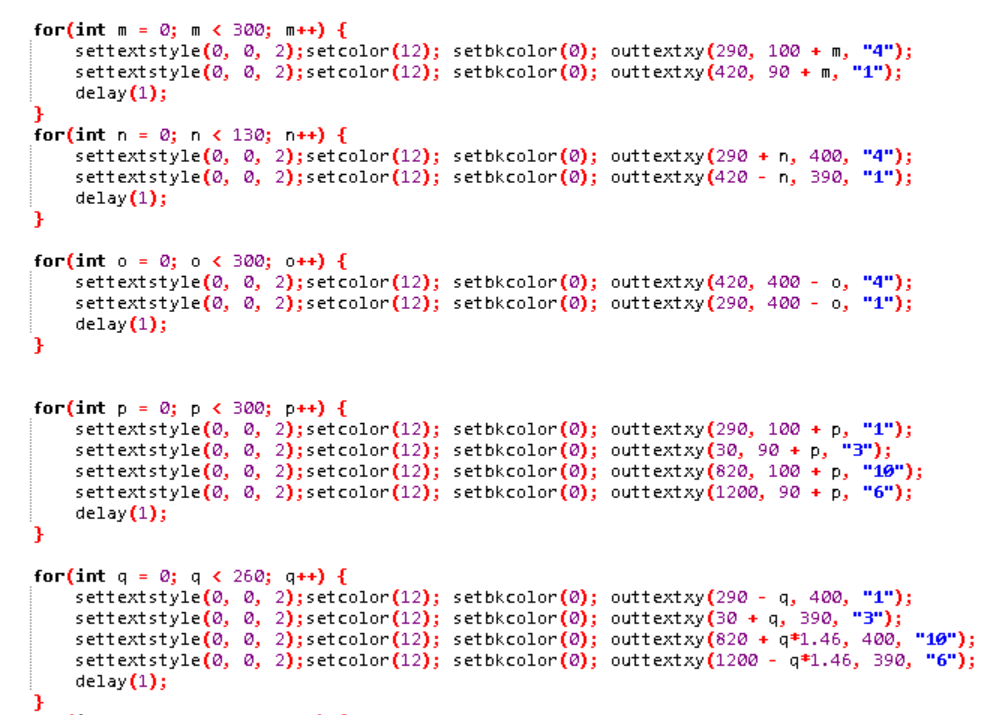
****

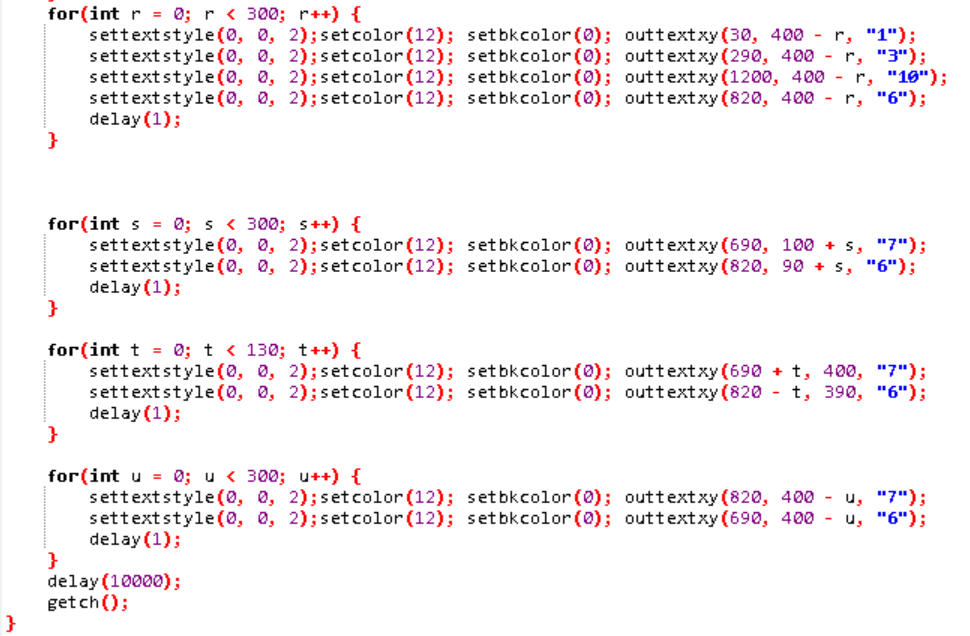
**Chương trình chính**





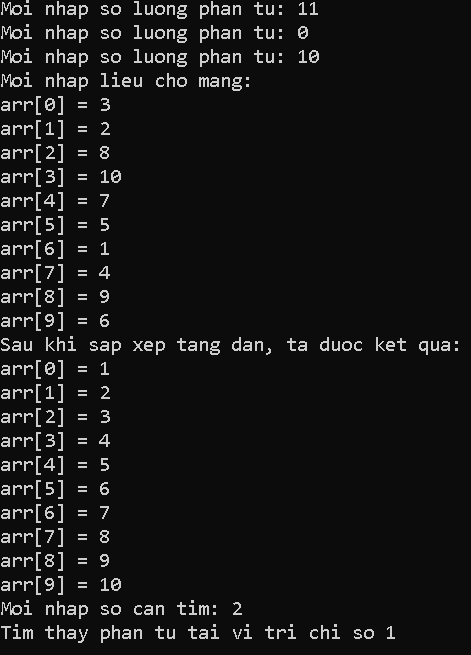


****

****

# CHƯƠNG 3. KẾT QUẢ THỰC HIỆN

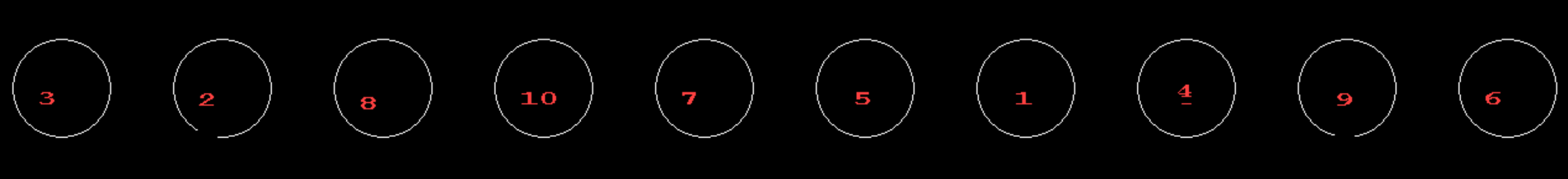
## *3.1. Hình ảnh thuật toán tìm kiếm nhị phân trong mảng*



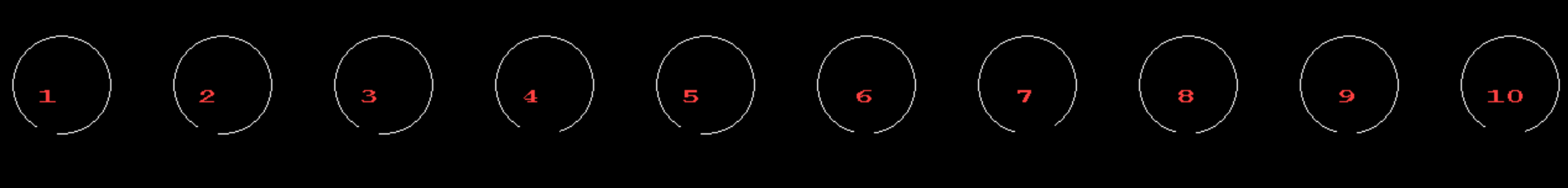
Hình 3.1 Kết quả của thuật toán tìm kiếm nhị phân trong mảng

Từ hình 3.1 ta thấy giới hạn của phần tử nhập là 1 đến 10, sau khi nhập đúng 10 phần tử ngẫu nhiên thì chương trình sẽ tự sắp xếp và sau đó người dùng sẽ chọn 1 vị trí bất kì và mảng sẽ đưa ra vị trí con số đang ở phù hợp.

## *3.2. Hình ảnh mảng chưa sắp xếp trong thư viện graphics.h*



Hình 3.2 Hình ảnh mảng chưa sắp xếp trong thư viện graphics.h



Hình 3.3 Hình ảnh mảng đã được sắp xếp trong thư viện graphics.h



Hình 3.4 Họ tên sinh viên thực hiện và giáo viên hướng dẫn

Từ hình 3.2 ta thấy thuật toán sẽ bắt từ 10 con số cho ngẫu nhiên và chưa được sắp xếp, sau đó hình 3.3 là những con số đã được sắp xếp và phía dưới là hình 3.4 là họ tên sinh viên và giáo viên hướng dẫn.

# THẢO LUẬN

Đề tài đã cài đặt được thuật toàn tìm kiếm nhị phân và nhập dữ liệu từ bàn phím. Ngoài ra còn tìm hiểu thêm về thuật toán bảng băm. Thuật toán tìm kiếm nhị phân chạy nhanh hơn tìm kiếm tuần tự nhưng cũng có một số nhược điểm. Tìm kiếm nhị phân có thể chậm hơn bảng băm. Nếu nội dung danh sách bị thay đổi thì danh sách phải được sắp xếp lại trước khi sử dụng tìm kiếm nhị phân. Thao tác này thường tốn nhiều thời gian.

Tuy nhiên, đề tài vẫn chưa cài đặt được thuật toán tìm kiếm khi sử dụng thư viện graphics.h , chỉ mới sắp xếp được mảng tăng dần.

Trong tương lai, em sẽ tiếp tục tìm hiểu thêm và cài đặt thuật toán hoàn thiện hơn nữa. Thuật toán tìm kiếm nhị phân là một thuật toán cơ bản nhưng được sử dụng rất nhiều trong lập trình thi đấu và trong cuộc sống. Việc áp dụng một cách khéo léo sẽ giúp bạn giải quyết và cải thiện các bài toán lập trình của mình.

# 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