MÔN CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

BÁO CÁO ĐỒ ÁN CUỐI KÌ

XÂY DỰNG CÂY GIA PHẢ (TREE)

# LỜI CẢM ƠN

Để hoàn thành đồ án này, trong lúc tìm hiểu và thu thập nhóm thực hiện đã nhận được sự giúp đỡ từ giảng viên và các bạn. Đặc biệt, đối với giảng viên ThS. Trần Công Tú, thầy đã hướng dẫn giúp đỡ trong quá trình làm đề tài.

Trong lúc thực hiện, còn vướng phải nhiều khó khăn sai phạm từ nhóm thực hiện, mong thầy và các bạn góp ý để giúp tối thiểu hóa các hạn chế, qua đó có thêm tư liệu mới để học tập cũng như xây dựng chương trình sau này.

Nhóm thực hiện xin chân thành cảm ơn!

**MỤC LỤC**

[LỜI CẢM ƠN 2](#_Toc27048396)

[DANH MỤC CÁC HÌNH 4](#_Toc27048397)

[DANH MỤC CÁC BẢNG 4](#_Toc27048398)

[PHẦN MỞ ĐẦU 5](#_Toc27048399)

[1. Lí do chọn đề tài: 5](#_Toc27048400)

[2. Mục tiêu cần đạt: 5](#_Toc27048401)

[3. Phương pháp nghiên cứu: 5](#_Toc27048402)

[4. Giới thiệu về cây gia phả: 5](#_Toc27048403)

[5. Giới thiệu về đề tài 6](#_Toc27048404)

[PHẦN NỘI DUNG 7](#_Toc27048405)

[1. Cấu trúc dữ liệu được sử dụng trong chương trình: 7](#_Toc27048406)

[1.1. Giới thiệu sơ lược về cấu trúc cây: 7](#_Toc27048407)

[1.2. Biểu diễn cây: 8](#_Toc27048408)

[1.3. Phương pháp duyệt cây: 9](#_Toc27048409)

[1.4. Các hàm tương tác trên cây: 9](#_Toc27048410)

[2. Chạy chương trình: 10](#_Toc27048411)

[2.1. Thiết kế giao diện chương trình: 10](#_Toc27048412)

[2.2. Kiểm thử và chạy debug: 13](#_Toc27048413)

[PHÂN CÔNG CÔNG VIỆC: 14](#_Toc27048414)

[KẾT LUẬN 15](#_Toc27048415)

[PHỤ LỤC 15](#_Toc27048416)

[TÀI LIỆU THAM KHẢO 15](#_Toc27048417)

# DANH MỤC CÁC HÌNH

*Hình 1: Cây gia phả (nguồn: PNGTREE)*

*Hình 2: Sơ đồ cây lưu trữ một gia phả*

*Hình 3: Cây nhị phân*

*Hình 4: Định nghĩa Struct GiaPha để tạo cấu trúc cây theo kiểu mảng*

*Hình 5: Hàm tạo gia phả*

*Hình 6: Hàm thêm người*

*Hình 7: Giao diện màn hình chính*

*Hình 8: Nhập tên người (khi cây rỗng)*

*Hình 9: Nhập tên người (khi đã có nút gốc)*

*Hình 10: Gia phả in dưới dạng danh sách*

*Hình 11: Sơ đồ gia phả*

*Hình 12: Tìm kiếm người trong gia phả*

# 

# DANH MỤC CÁC BẢNG

*Bảng 1: Kiểm thử và Debug*

*Bảng 2: Phân công công việc*

# PHẦN MỞ ĐẦU

## Lí do chọn đề tài:

Với đề tài “xây dựng cây gia phả” có thể giúp người sử dụng vận dụng các kiến thức của mình để thực hiện việc xây dựng một cây lưu giữ thông tin của tổ tiên. Ngoài ra, việc tạo ra một cây lưu giữ thông tin giúp cho người sử dụng hiểu biết về nguồn cội của mình, tìm hiểu về thông tin dòng họ gần gũi với gia đình mà không cần lập mô hình thực tế mà còn đảm bảo tính chính xác.

## Mục tiêu cần đạt:

Hoàn thành được sản phẩm, hỗ trợ người sử dụng thực hiện các thao tác tương tác lên sơ đồ gia phả (hiển thị thông tin gia phả, thêm thành viên vào gia phả, tìm kiếm và hiển thị người trong gia phả). Bản thân tác giả hiểu được, và áp dụng đề tài cho các vấn đề liên quan.

## Phương pháp nghiên cứu:

Vận dụng các kiến thúc được học từ bộ môn “Cấu trúc dữ liệu và giải thuật”, tham khảo các tài liệu, giáo trình được cung cấp. Ngoài ra vận dụng thêm nguồn kiến thức từ Internet hỗ trợ cho việc hoàn thành đồ án được thành công nhất.

## Giới thiệu về cây gia phả:



*Hình 1: Cây gia phả (Nguồn: PNGTREE)*

Gia phả hay gia phổ là một bản ghi chép đầy đủ với các thông tin là họ và tên, tuổi, ngày giỗ, vai trò, công đức của người trong dòng họ. Có thể nói một cách khác thì cây gia phả là một bản sử kí lưu trữ thông tin của thế hệ đi trước lưu truyền từ đời này sang đời khác.

Từ thời phong kiến tại các nước Đông á, chịu ảnh hưởng của Đông Á, cây gia phả được các thế hệ trong vương triều ghi lại thông tin của đời trước như là một cách để giữ đạo Trung đạo Hiếu và ghi nhớ công ơn của tổ tiên, những người có công dựng nước và giữ nước.

## Giới thiệu về đề tài

Chương trình được xây dựng cho đồ án là chương trình cây được cấu trúc như cây đa phân. Sản phẩm cuối cùng là cây lưu trữ được biểu diễn để thực hiện các chức năng nhầm hỗ trợ người sử dụng.

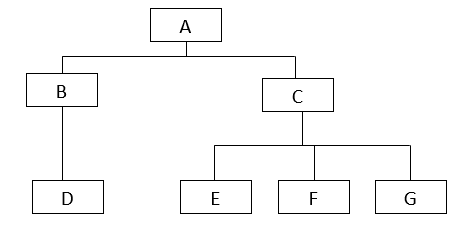
Trong chương trình sẽ có các hàm sau:

* Themnguoi(): giúp người sử dụng có thể thêm thông tin lưu trữ vào trong cây. Thông tin sẽ được nhập từ bàn phím.
* Hienthi(): in ra thông tin mà cây đang lưu trữ
* Ingiapha(): in danh sách người trong gia
* Timkiem(): giúp người sử dụng tra cứu thông tin cần tìm trong gia phả.

# PHẦN NỘI DUNG

## Cấu trúc dữ liệu được sử dụng trong chương trình:

* Để giải quyết bài toán một cây gia phả lưu trữ các thông tin cá nhân của thành viên trong dong họ ta có thể sử dụng cấu trúc cây đa phân n để gia quyết. Đối với kiểu dữ liệu vào là thông tin cá nhân hỗn hợp thì sử dụng cấu trúc này là hợp lí nhất để thể hiện tính minh bạch, rõ ràng. Tuy nhiên, điều này lại gặp trở ngại là bậc n phụ thuộc vào số con của một thành viên nên khó mà xác nhận được số con n trong lý thuyết.
* Với gốc cây lưu giữ thông tin của thế hệ trước, nhánh con là thông tin cá nhân của các thế hệ liền kề phía sau.



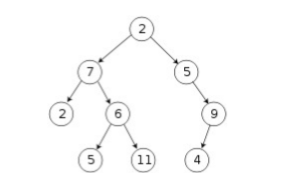
*Hình 2: Sơ đồ cây lưu trữ một gia phả*

### Giới thiệu sơ lược về cấu trúc cây:

Trong cấu trúc dữ liệu và giải thuật, cây là một tập hợp T các phần tử (gọi là nút của cây) trong đó có một nút đặc biệt gọi là nút gốc, các nút còn lại được chia thành những tập rời nhau T\_1, T\_2, …, T\_n theo mối quan hệ phân cấp trong đó T\_i cũng là một cây. Mỗi nút ở cấp I sẽ quản lí các nút ở i+1. Quan hệ này còn được gọi là quan hệ cha–con. [2]

Cây trong cấu trúc dữ liệu đầu tiên là mô phỏng (hay nói cách khác là sự sao chép) của cây (có gốc) trong lý thuyết đồ thị. Hầu như mọi khái niệm trong cây của lý thuyết đồ thị đều được thể hiện trong cấu trúc dữ liệu. Tuy nhiên cây trong cấu trúc dữ liệu được ứng dụng phong phú và hiệu quả hơn trong nhiều giải thuật. Khi phân tích các giải thuật trên cấu trúc dữ liệu cây, người ta vẫn thường vẽ ra các cây tương ứng trong lý thuyết đồ thị. [1]

Cây trong cấu trúc dữ liệu được phân loại dựa trên số nút (Node) mà nó đang lưu trữ. Nếu số nút của nó không quá hai thì nó sẽ được gọi là cây nhị phân, nếu lớn hơn thì nó sẽ được gọi là cây tổng quát hay còn tên khác là cây đa phân. Để biểu diễn cây gia phả trong đồ án này thì chúng em đã sử dụng cây tổng quát lưu trữ thông tin với mỗi bậc của cây ứng với từng thế hệ, bắt đầu với nút gốc có bậc bằng 0 (ứng với thế hệ 0). Với lí do chính khi chọn như vậy vì cây nhị phân không biểu diễn các thế hệ trên cùng một hang, như vậy thì khi in sơ đồ sẽ khó xác định được thế hệ của tựng người trong dòng họ

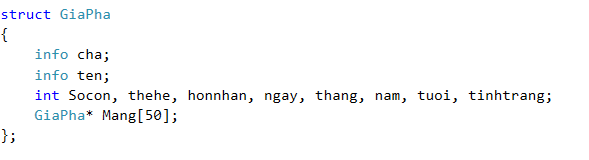


*Hình 3: Cây nhị phân*

### Biểu diễn cây:

Cây trong đề tài được xây dựng dựa trên nền tảng xây dựng mảng một chiều.

Để xây dựng cây thì đầu tiên ta định nghĩa hàm Struct Gia nút gốc chứa các thông tin cá nhân sau đó dùng đệ quy gọi lại hàm struct để tạo ra nút con lưu trữ các thông tin của thế hệ sau, thông tin ấy sẽ được lưu vào một mảng kiểu struct với tên mang[n] với n là số con max.



*Hình 4:Định nghĩa Struct GiaPha để tạo cấu trúc cây theo kiểu mảng*

Sau khi định nghĩa Struct GiaPha, ta dùng lệnh typedef và định nghĩa cho kiểu Tree: typedef GiaPha\* Tree;. Từ đây ta đã có kiểu Tree hỗ trợ cho hàm TaoGiaPha();



*Hình 5: Hàm tạo gia phả*

*(Hình 5)*: Tạo ra một biến Tree và gán tên là cây, ta dùng con trỏ để lấy trỏ đến cùng nhớ của các thông tin và sao lưu thông tin ấy vào biến cây, trong hàm Tree đó t còn gán thếm số con của nút là i để tạo ra mảng lưu trữ với các phần tử (phần tử con) sẽ có kiểu là Tree.

Ý tưởng chính: tạo ra nút gốc và cho liên kết với một mảng chứa thông tin các thành phần con, trong mảng con ấy thì từng phần tử sẽ được liên kết với mảng chứa thông tin của thành phần con khác.

### Phương pháp duyệt cây:

Duyệt tuần tự từ trái sang phải các phần tử của cây theo biến “thehe” của cây.

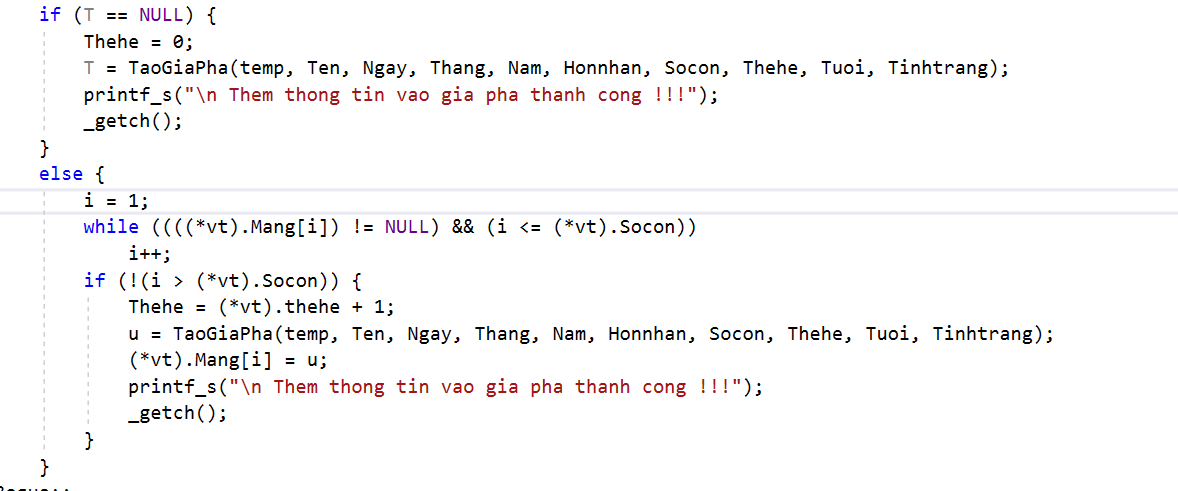
### Các hàm tương tác trên cây:

* **Hàm Thêm Người:**

Ở hàm thêm người, đầu tiên khởi tạo cây rỗng và nhập thông tin của nút gốc (tổ tiên đầu) và găn với thế hệ đầu tiên. Sau đó dựa theo số con mà nút gốc có được khởi tạo 1 biến i với giá trị ban đầu bằng 1. Dựa theo số con tăng i bằng với số con. Nút gốc sẽ chứa 1 mảng gồm 50 phần tử thể hiện số con mà nút đó có được, mỗi số con là 1 phần tử mảng được lưu thông tin bằng cây. Tiếp đến cho các thế hệ tiếp theo.

**\*Hàm Thêm Người có hai trường hợp chính xảy ra:**

* Nếu cây ban đầu rỗng (Tree T==NULL) thì hàm này sẽ trở thành hàm nhập bình thường và sẽ có một dòng là nhập số con để tạo mảng lưu trữ nút con (nếu có).



*Hình 6: Hàm thêm người*

* Nếu cây đã có gốc (Tree T != NULL) thì hàm sẽ cho ra dòng lệnh nhập là “Cha mẹ của người đó” dung để liên kết với nút gốc của nó. Tùy vào số con được khai báo ở nút gốc thì ta có thể liên kết bao nhiêu con với nút gốc đó.
* **Hàm Tìm Người:**

Hàm tìm thông tin 1 người được thực hiện trước hết thông qua hàm tìm kiếm tên (tùy ông có nên thêm vào không). Nếu tìm được tên trong gia phả sẽ truy xuất 1 số thông tin đã nhập ở hàm thêm người lên màn hình, ở phần xuất thông tin sẽ có 2 hàm phụ là hàm tìm kiếm thế hệ (xuất 1 số thông tin những người cùng thế hệ) và hàm in con (xuất thông tin con của người này).

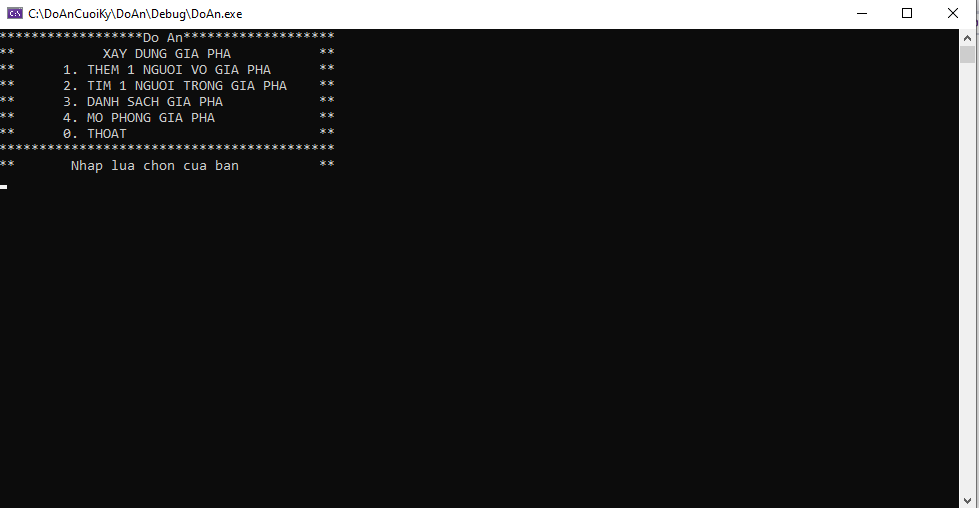
* **Hàm in danh sách:**

Hàm in danh sách thể dùng đệ quy để in một số thông tin của tất cả những người đã nhập lên trên kể cả nút gốc

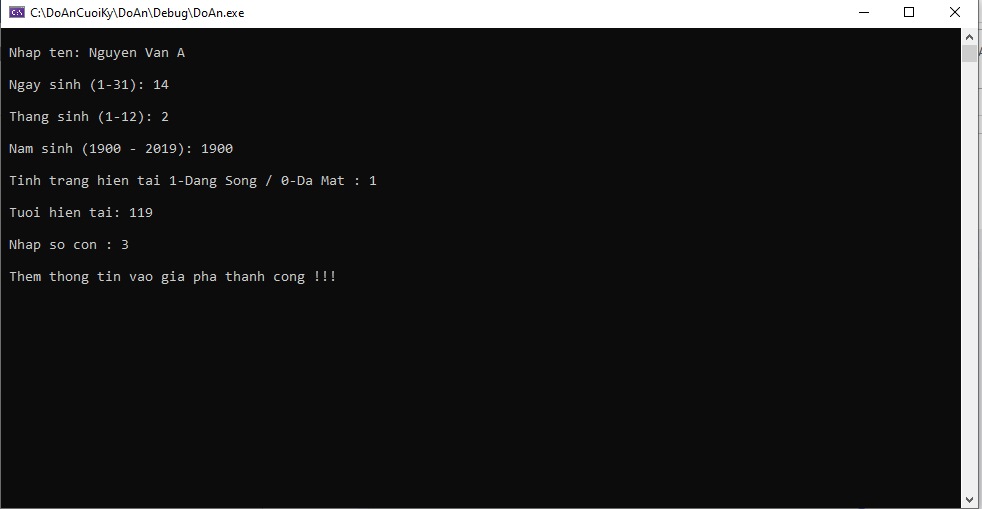
## Chạy chương trình:

### Thiết kế giao diện chương trình:

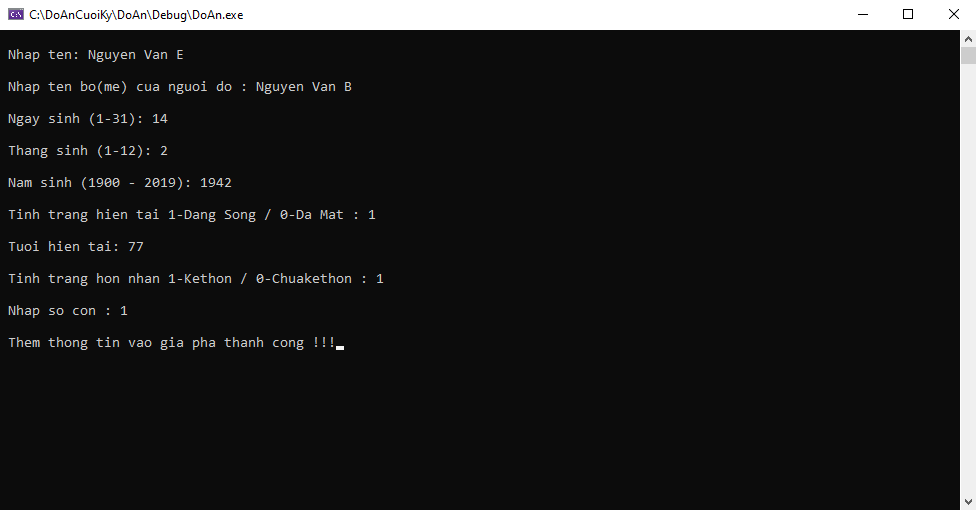
Được thiết kế trên giao diện Console:

****

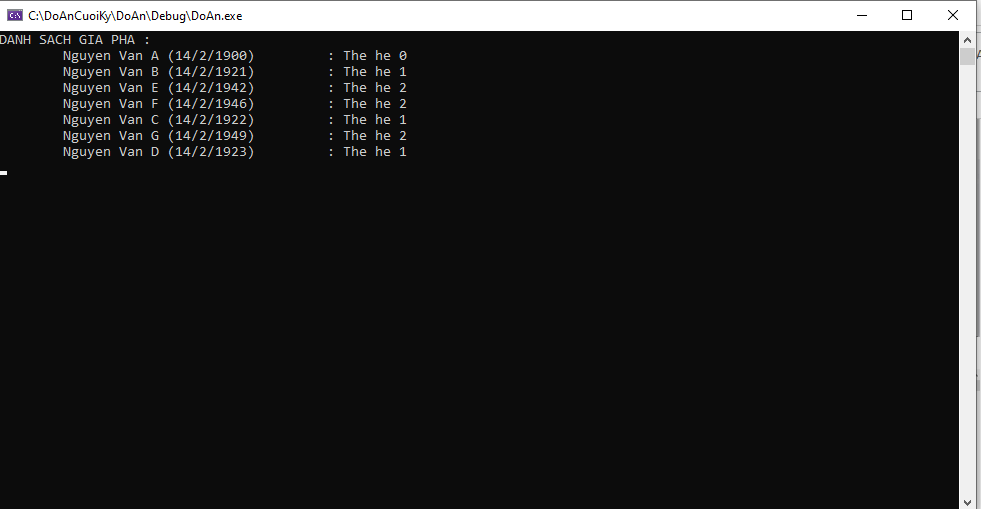
*Hình 7: Giao diện màn hình chính*



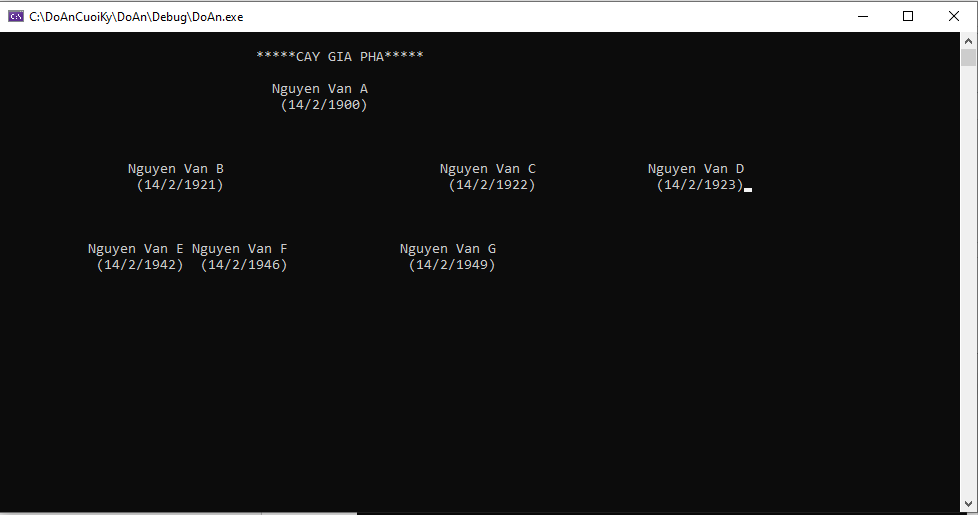
*Hình 8: Nhập tên người (khi cây rỗng)*



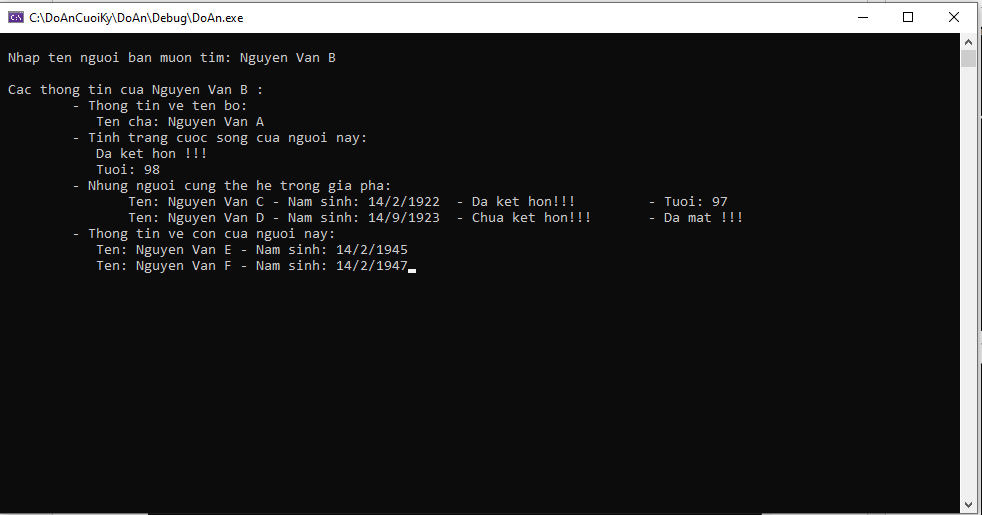
*Hình 9: Nhập tên người (khi đã có nút gốc)*



*Hình 10: Gia phả in dưới dạng danh sách*



*Hình 11: Sơ đồ gia phả*



*Hình 12: Tìm kiếm người trong gia phả*

### Kiểm thử và chạy debug:

Chạy thử với mẫu là một nút gốc và 2 nút con (1, 2) của nó. Sau khi sử dụng hàm mô phỏng cây xuất hiện lỗi chỉ hiển thị nút gốc và nút con 2. Tìm hiểu nguyên nhân khi chạy debug thì con 2 chèn lên con 1 chứ không phải bị mất.

Sửa lỗi: Cấp phát riêng bộ nhớ cho từng con để tăng khoảng trắng giữa 2 con.

# PHÂN CÔNG CÔNG VIỆC:

*Bảng 2: Bảng phân công công việc*

# KẾT LUẬN

* **Ưu điểm**
* Giúp người dùng dễ dàng lưu trữ thông tin dòng họ
* Giao diện thiết kế dễ sử dụng, dễ nhìn, dễ tiếp cận
* **Nhược điểm:**
* Tốn khá nhiều dung lượng bộ nhớ.
* Chưa lưu trữ được cụ thể rõ ràng hơn thông tin gia đình.
* Sản phẩm còn một vài lỗi nhỏ chưa khắc phục được
* Thế hệ lưu trữ tối đa chỉ có 5
* **Phương hướng phát triển:**
* Cải tiến thuật toán để chương trình chạy nhanh, ổn định hơn.
* Cải tiến thiết kế đồ họa cho dễ nhìn và dễ hiểu hơn.
* Xây dựng chương trình dưới dạng đối tượng (object) với các thuộc tính và phương thức rõ rang để khắc phục nhiều vấn đề như thế hệ, …

# PHỤ LỤC

**Link GitHub:** <https://github.com/nguyenminhthuan2309/do-an-cay-gia-pha>

# TÀI LIỆU THAM KHẢO

**[1]:** <https://vi.wikipedia.org/wiki/Gia_ph%E1%BA%A3>

**[2]: Sách Cấu Trúc dữ liệu và giải thuật ( BS: ThS. Trần Hạnh Nhi & ThS. Dương Anh Đức; Hiệu đính: GS. TS. Hoàng Kiếm).**

**[3]:** <https://voer.edu.vn/c/cai-dat-cay/3bed1b9f/e62e9079>