**ĐẠI HỌC QUỐC GIA TP HCM**

**ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**

**🙢🕮🙠**



**ĐỒ ÁN CUỐI KÌ  
GAME CỜ CARO TRÊN GIAO DIỆN CONSOLE**

**Môn học :** Kỹ Thuật Lập Trình

**Giáo viên hướng dẫn :**  Lê Ngọc Thành

**Danh sách sinh viên thực hiện :**

* Trần Phạm Khánh Sơn MSSV: 1712733

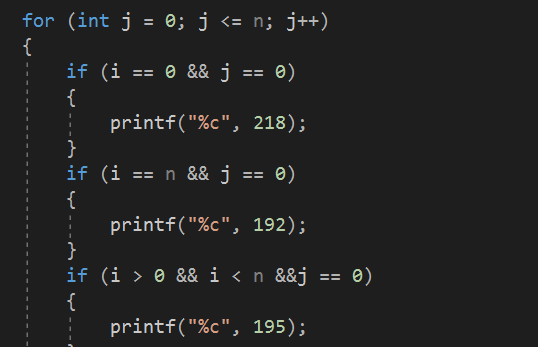
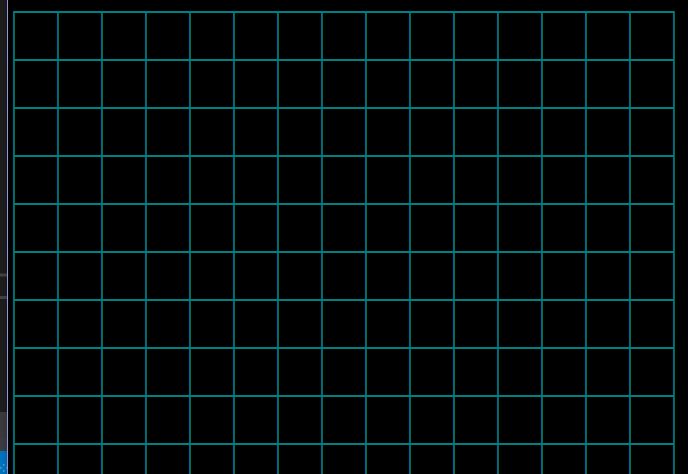
Hồ Chí Minh, tháng 5 năm 2018

1. **Mô tả yêu cầu đề bài:**

* Xây dựng game cờ caro với những chức năng cơ bản trên giao diện Console.
* Luật chơi: Hai bên lần lượt thay phiên nhau đặt những nước đi bằng các dấu X, O trên bàn cờ. Bên nào có năm quân liền nhau mà không có quân khác chặn liền hai đầu trên một trong các hàng dọc, hàng ngang hay đường chéo là giành chiến thắng.
* Một số yêu cầu cụ thể của game:
* Giúp đỡ ( Help): Hiển thị thông tin hướng dẫn cách chơi và luật chơi.
* Tùy chọn (Options): Thay đổi các thông số của game, lựa chọn chế độ chơi và chọn ra quân trước.
* Game mới (New Game): Bắt đầu ván chơi với các thông số đã đặt.
* Lưu game (Save Game): Cho phép lưu lại ván cờ đang chơi, lượt đánh tiếp theo ( O hay X).
* Tải game (Load Game): Cho phép tải lại ván cờ đã lưu để chơi tiếp.
* Thông tin (About): Cho phép hiển thị thông tin của nhóm tác giả, phiên bản trò chơi.
* Thoát (Exit): Thoát khỏi chương trình.

1. **Mô tả các hàm xử lý chính:**

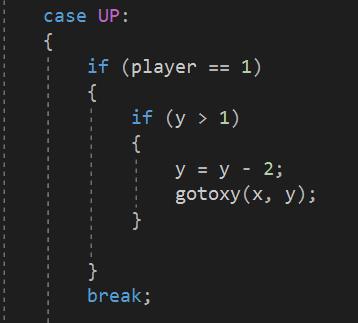
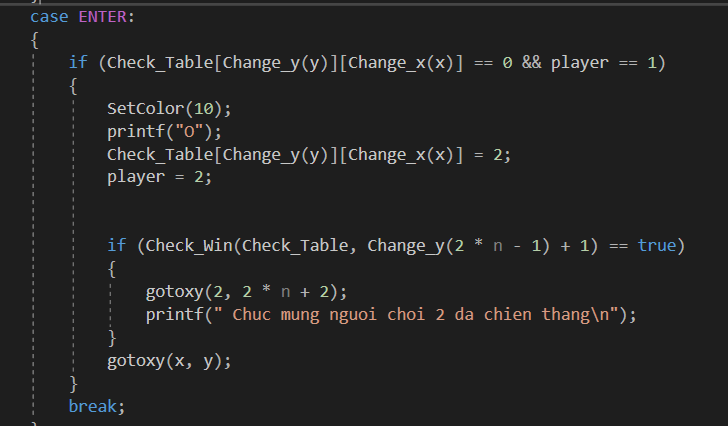
* **Vẽ bàn cờ**:

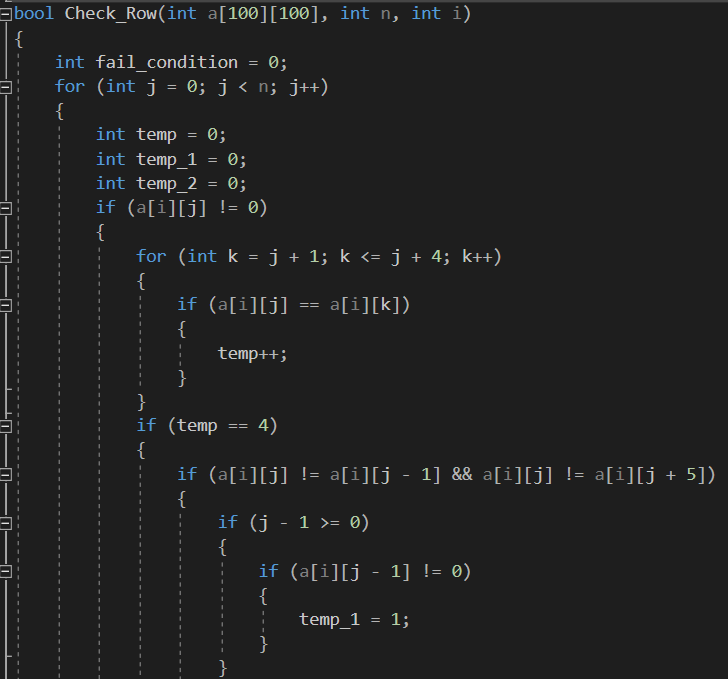
Hàm có thông số đầu vào là size của bàn cờ, ở đây đã cố định là 15\*15 lần lượt dùng các mã ASCII của một số kí tự phù hợp để xây dựng lên bàn cờ caro như mong muốn bằng cách in ra các kí tự đó ở các vị trí thích hợp bằng vòng lặp. Bên trên là minh họa một đoạn của hàm vẽ bàn cờ.

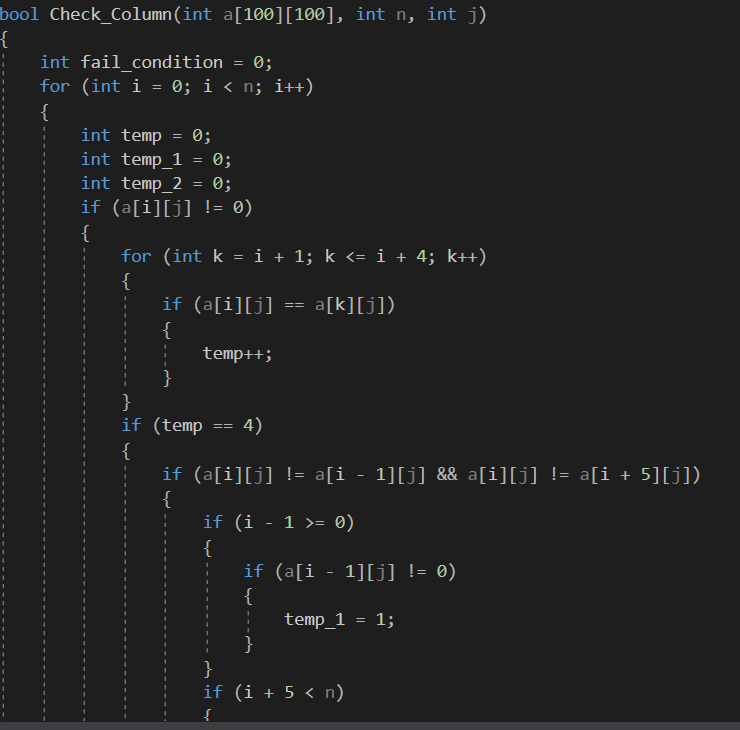
Sau khi chạy hàm vẽ bàn cờ cùng các bước thay đổi màu hiển thị trên giao diện console sẽ có kết quả như hình minh họa bên phải đồng thời khởi tạo một mảng hai chiều số nguyên để lưu trạng thái của từng ô trong bàn cờ, ban đầu mặc định là 0 tức là chưa có quân cờ, sau khi người chơi thứ 1 đặt quân cờ vào ô nào ô tương ứng trong mảng hai chiều sẽ thay đổi từ 0 sang 1 và tương tự như vậy với người chơi thứ 2.

* **Xử lý các tính năng di chuyển và đặt quân cờ của người chơi:**

Việc này được thực hiện dựa trên hàm chính là getch() hay \_getch() trong thư viện stdio.h chuẩn của C để bắt phím được nhập vào từ bàn phím có đúng với phím đã được định sẵn trước để xử lý dựa trên vòng lặp while cũng như cấu trúc switch – case. Gồm hai phần chính là di chuyển và đặt quân cờ cụ thể như sau:

* **Đối với xử lý di chuyển:** đầu tiên khởi tạo biến player đánh đánh dấu lượt của mỗi người chơi – mặc định được đặt là 1 tức là người chơi 1 có quyền đi. Nếu phím di chuyển được bắt qua hàm \_getch() cũng như thông số của biến player phù hợp với phím di chuyển đã được định sẵn sẽ xử lý chạy hàm hỗ trợ gotoxy(int x, int y) làm thay đổi tọa độ của con trỏ đặt quân cờ theo ý muốn của người dùng. Minh họa bên trên là phím di chuyển lên trên bằng mũi tên hướng lên của người chơi 1.
* **Đối với xử lý đặt quân cờ:** Ngoài việc biến player phải tương thích với lượt đúng của người chơi để nhấn phím enter hay spacebar để đặt quân cờ thì việc xử lý đặt quân cờ còn thêm một bước là kiểm tra xem ô được chỉ định được quân cờ trong mảng số nguyên hai chiều Check\_Table có đang ở trạng thái trống (tức là bằng 0) hay không. Nếu có hàm sẽ đổi màu quân cờ tương ứng của người chơi rồi in ra màn hình kí hiệu quân cờ X hay O của người chơi tại vị trí đã định sẵn đồng thời thay đổi ô tương ứng trong mảng Check\_Table về giá trị tương ứng của người chơi đánh dấu là ô đã cho đã có lượt đánh. Ngoài ra mỗi khi việc xử lý đặt quân cờ được thực hiện còn kiểm tra việc thắng thua của người chơi qua hàm Check\_Win nếu hàm trả về kết quả đúng tức là người chơi đã dành chiến thắng thì in ra thông báo đồng thời dừng chương trình.
* **Xử lý kiểm tra chiến thắng:**

Việc xử lý kiểm tra sự chiến thắng trong trò chơi được thực hiện qua hàm Check\_Win với 6 hàm con hỗ trợ lần lượt là:  
- Check\_Row(): Thực hiện việc kiểm tra theo hàng mỗi khi quân cờ được đặt xuống có tham số đầu vào là mảng Check\_Table lưu trạng thái quân cờ trên bàn cờ, kích thước của bàn cờ và vị trí đặt quân cờ cần kiêm tra. Hàm sẽ tìm kiếm trên hàng ngang xem có 5 quân cờ cùng một kiểu nằm cùng nhau mà không bị chặn hai đầu hay không bằng cách duyệt mảng Check\_Table. Hàm trả về true nếu tìm được và ngược lại thì false.

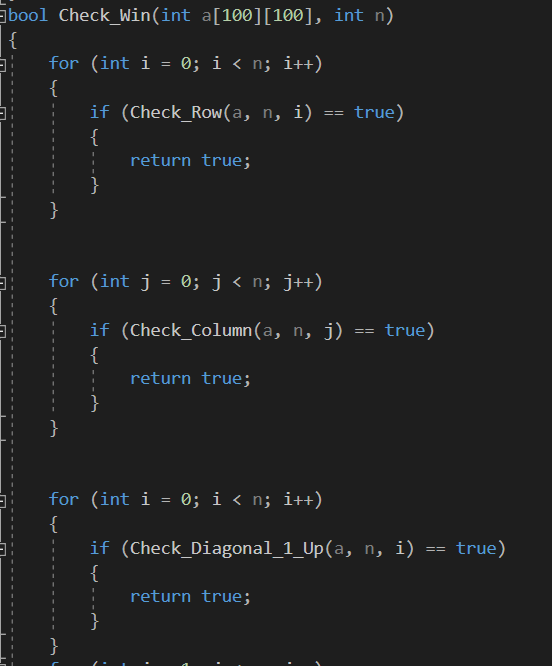
- Check\_Column(): Thực hiện việc kiểm tra theo cột mỗi khi quân cờ được đặt xuống có tham số đầu vào là mảng Check\_Table lưu trạng thái quân cờ trên bàn cờ, kích thước của bàn cờ và vị trí đặt quân cờ cần kiêm tra. Hàm sẽ tìm kiếm trên cột dọc xem có 5 quân cờ cùng một kiểu nằm cùng nhau mà không bị chặn hai đầu hay không bằng cách duyệt mảng Check\_Table. Hàm trả về true nếu tìm được và ngược lại thì false.

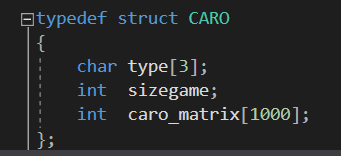
- Check\_Diagonal\_1\_Up(): Thực hiện việc kiểm tra theo đường chéo trên phía trái mỗi khi quân cờ được đặt xuống có tham số đầu vào là mảng Check\_Table lưu trạng thái quân cờ trên bàn cờ, kích thước của bàn cờ và vị trí đặt quân cờ cần kiêm tra. Hàm sẽ tìm kiếm trên đường chéo trên phía trái xem có 5 quân cờ cùng một kiểu nằm cùng nhau mà không bị chặn hai đầu hay không bằng cách duyệt mảng Check\_Table. Hàm trả về true nếu tìm được và ngược lại thì false.

- Check\_Diagonal\_1\_Down() : Thực hiện việc kiểm tra theo đường chéo trên phía phải mỗi khi quân cờ được đặt xuống có tham số đầu vào là mảng Check\_Table lưu trạng thái quân cờ trên bàn cờ, kích thước của bàn cờ và vị trí đặt quân cờ cần kiêm tra. Hàm sẽ tìm kiếm trên đường chéo trên phía phải xem có 5 quân cờ cùng một kiểu nằm cùng nhau mà không bị chặn hai đầu hay không bằng cách duyệt mảng Check\_Table. Hàm trả về true nếu tìm được và ngược lại thì false.

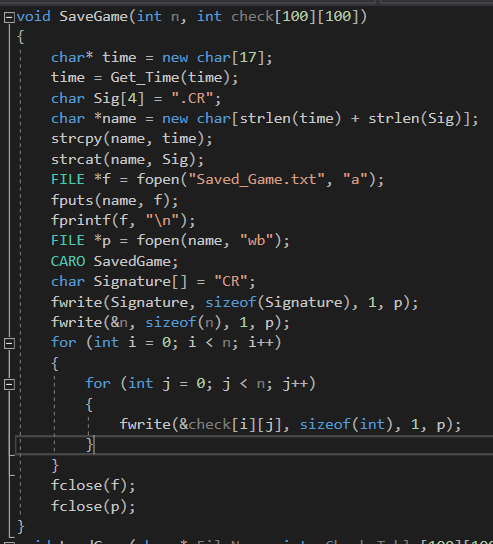
- Check\_Diagonal\_2\_Up(): Thực hiện việc kiểm tra theo đường chéo dưới phía phải mỗi khi quân cờ được đặt xuống có tham số đầu vào là mảng Check\_Table lưu trạng thái quân cờ trên bàn cờ, kích thước của bàn cờ và vị trí đặt quân cờ cần kiêm tra. Hàm sẽ tìm kiếm trên đường dưới trên phía phải xem có 5 quân cờ cùng một kiểu nằm cùng nhau mà không bị chặn hai đầu hay không bằng cách duyệt mảng Check\_Table. Hàm trả về true nếu tìm được và ngược lại thì false.

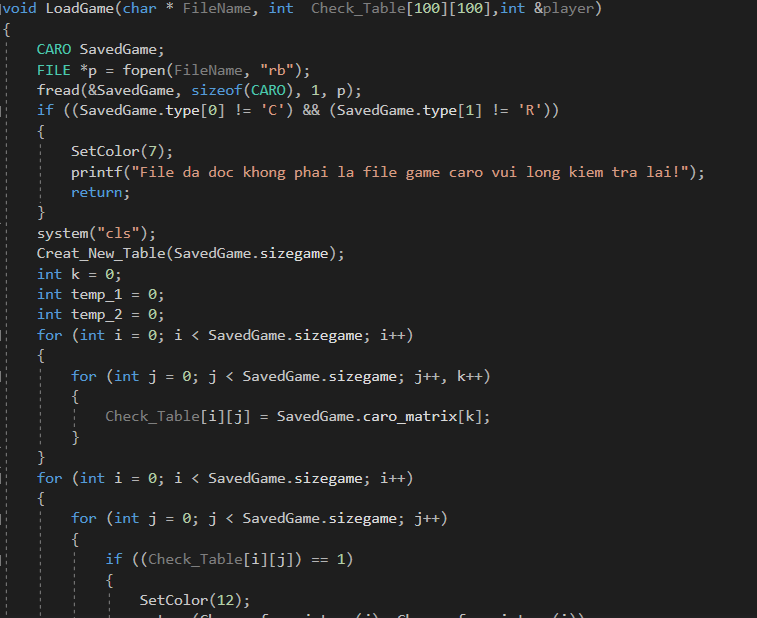
- Check\_Diagonal\_2\_Down(): Thực hiện việc kiểm tra theo đường chéo dưới phía trái mỗi khi quân cờ được đặt xuống có tham số đầu vào là mảng Check\_Table lưu trạng thái quân cờ trên bàn cờ, kích thước của bàn cờ và vị trí đặt quân cờ cần kiêm tra. Hàm sẽ tìm kiếm trên đường chéo dưới phía trái xem có 5 quân cờ cùng một kiểu nằm cùng nhau mà không bị chặn hai đầu hay không bằng cách duyệt mảng Check\_Table. Hàm trả về true nếu tìm được và ngược lại thì false.

- Hàm Check\_Win sẽ trả về kết quả đúng nếu như trong vòng lặp duyệt các hàm con có một hàm trả về kết quả đúng và ngược lại nếu cả 6 hàm con đều trả về kết quả là sai thì hàm Check\_Win sẽ trả về kết quả là sai.

* **Lưu game:**

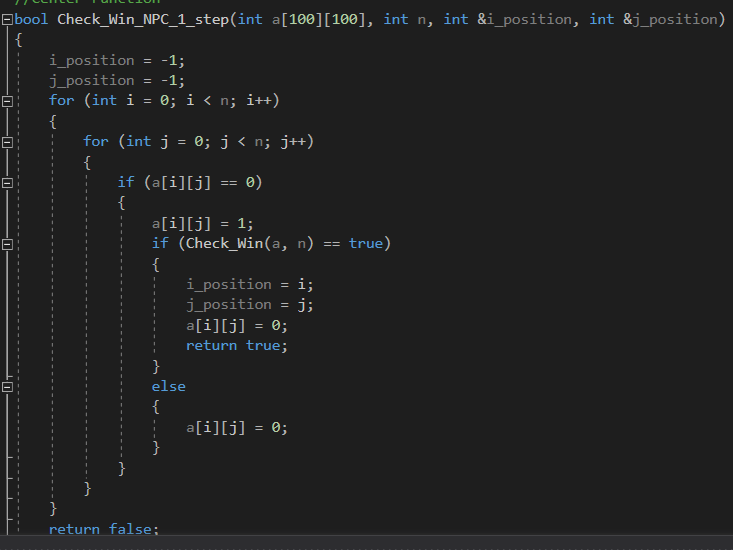
Ý tưởng của việc lưu game là đọc ghi cấu trúc tự định nghĩa vào file nhị phân như tập tin BMP. Để lưu được game trước tiên khởi tạo struct như sau type là chuỗi kí tự “CR” là chữ kí của file nhằm nhận dạng file trong lúc load. Số nguyên size game chỉ kích thước của game mặc định là 15\*15. Và cuối cùng quan trọng nhất là mảng caro\_matrix chứa thông tin của mảng số nguyên Check\_Table ở trên để lưu trạng thái của ván cờ caro

Hàm SaveGame sau khi thực hiện việc lấy thời gian hệ thống đã được gán trong chuỗi do hàm hỗ trợ Get\_Time thực hiện mở file với tên đã được định sẵn dưới dạng ghi nhị phân để ghi cấu trúc struct CARO cần ghi vào file. Hoàn tất việc lưu lại game chuẩn bị cho việc load game. Đồng thời cũng mở file SavedGame.txt dưới dạng thêm văn bản để thêm tên của file đã lưu vào tiện cho việc hiển thị danh sách các file đã lưu cho phần loadgame.

* **Load Game:**

Hàm LoadGame thực hiện việc đọc một file được nhập tên từ bàn phím rồi hiển thị ra màn hình màn chơi cho phép người chơi chơi tiếp. Khi người chơi nhấn phím F4 tức loadgame chương trình sẽ mở file SaveGame.txt đã tạo ở hàm save game đọc tất cả tên trong tập tin và hiển thị ra màn hình, Người dùng nhập tên file muốn load khi đó hàm loadgame thực hiện mở file vừa nhập tên rồi đọc struct đã lưu một cách trọn vẹn. Nếu kí tự chữ kí không phải là CR hàm sẽ đưa ra thông báo rồi kết thúc. Ngược lại hàm thực hiện các bước tiếp theo với việc đọc ma trận trạng thái bàn cờ rồi thực hiện đánh lại các ô đã được đánh trên bàn cờ đồng thời trong quá trình đánh hàm cũng đếm số ô đã đánh của mỗi người chơi Người chơi nào có số ô đã đánh ít hơn sẽ là người chơi đi trước sau khi load game. Nếu hai người có số ô đã đánh bằng nhau người chơi đi trước tính theo mặc định.

* **Chơi với máy:**

Ý tưởng là chia thành hai hướng để tiếp cận với ván đấu là tấn công và phòng ngự. Đầu tiên máy sẽ kiểm tra xem mình có thể thắng được trong một nước không bằng cách quét tất cả các ô trống còn lại trong bảng với thử cho từng ô trống giá trị.

Hàm kiểm tra máy có thể thắng trong một nước hay không?

Nếu tìm được ô trống cần tìm đánh dấu lại rồi thực hiện việc đặt quân cờ vào đó. Nếu không có máy sẽ chuyển qua bước phòng ngự bằng việc kiểm tra xem có ô nào người chơi có thể thắng trong một nước không với cách tương tự như tấn công trong một nước để chặn lại. Cứ như thế với tấn công thắng trong hai nước rồi phòng ngự người chơi thắng trong hai nước. Sang bước thứ ba không còn việc phòng ngự chuyển qua tấn công cứ như thế cho đến khi thắng.

1. **Đánh giá các chức năng đã thực hiện:**

|  |  |
| --- | --- |
| **Chức Năng** | **Đánh Giá** |
| Help | 100% |
| Options | 80% |
| New Game | 100% |
| Save Game | 100% |
| Load Game | 80% |
| About | 100% |
| Exit | 100% |

1. **Tài liệu tham khảo:**
2. <https://www.quora.com/What-is-gotoxy-function-used-in-c>
3. <http://www.cprogrammingnotes.com/question/tic-tac-toe-game.html>
4. <http://www.ee.bgu.ac.il/~microlab/MicroLab/Labs/ScanCodes.htm>
5. <https://www.epochconverter.com/programming/c>
6. https://www.geeksforgeeks.org/setcolor-function-c/