**PHƯƠNG PHÁP LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**13CTT11 – HKI\_2014-2015**

**Tên đồ án:**

**<CỜ VUA>**

**Sinh viên thực hiện:**

**<1012400 – Nguyễn Tấn Thành>**

**Thống kê công việc cá nhân:**

**<1012400 – 100%>**

**Sinh viên đánh giá kết quả thực hiện đồ án: 9.0 điểm**

1. **Thiết kế lớp:**

Ta có hai lớp chính là Bàn Cờ và Quân Cờ. Với lớp Bàn Cờ, ta lưu con trỏ hai chiều chỉ vị trí hàng và cột của bàn cờ. Tại mỗi vị trí ta lưu một quân cờ hoặc NULL nếu không có quân cờ nào tại vị trí đó.

Bàn cờ còn lưu số nước đã đi để giúp xác định bên nào (Trắng hay Đen) sẽ được đi tiếp theo.

Với lớp Quân Cờ, mỗi đối tượng sẽ có các thuộc tính là tên gì (Chốt, Xe, Mã, Tượng, Hậu và Vua), màu (trắng hay đen), còn sồng không, ở hàng nào, cột nào. 6 lớp thừa kế lớp Quân Cờ là Chốt, Xe, Mã, Tượng, Hậu và Vua. Mỗi loại Quân Cờ sẽ có các bước đi phù hợp được cài đặt trong phương thức move(int hangmoi, int cotmoi, QuanCo\* \*\*banco), trong đó hàng mới và cột mới là nơi đến của quân cờ và hàm move sẽ xác định bước đi có thích hợp không trong điều kiện của banco hiện tại.

Lớp Bàn Cờ sẽ cài đặt phương thức move(string tu, string den) có nhiệm vụ kiểm tra loại quân cờ tại điểm xuất phát (nếu có), sau đó xác định bước đi đến điểm đích có phù hợp không đối với quân cờ đó.

Sơ đồ lớp có thể nắm bắt được các diễn biến có thể xảy ra đối với một ván cờ. Tuy nhiên, hiện tại, các bước đi đặc biệt như phong Hậu hay kiểm tra thắng thua vẫn chưa được cài đặt.

1. **Tập tin dữ liệu**

* **Định dạng:**

Tập tin dữ liệu banco.txt có dạng

**\* Xet Tuongt \* \* \* \* \***

**Chott \* \* \* \* Mad \* \***

**\* Chott Mat Chott \* \* \* Xed**

**\* \* \* \* \* Chott \* \***

**\* \* Chotd \* \* Mad Chott Vuat**

**\* \* \* Chotd \* \* \* \***

**\* Tuongd \* Vuad Chotd Chotd \* \***

**\* \* \* \* \* \* \* \***

**64**

Trong đó tám hàng đầu tiên lưu bàn cờ. Tại vị trí khuyết quân cờ, ta lưu \*. Các vị trí có quân cờ, ta lưu tên và màu quân cờ, chẳng hạn, Xet: Xe trắng hay Mad: Mã đen.

Hàng cuối cùng lưu số nước cờ đã đi, ở đây là 64, do đó lượt đi tiếp theo thuộc về bên trắng.

* **Lưu dữ liệu:**

Bàn cờ được lưu từ hàng 0-7 và từ cột 0-7. Các cột cách nhau một khoảng trắng. Ngoài ra, số nước đi được lưu ở hàng 8.

* **Nạp dữ liệu:**

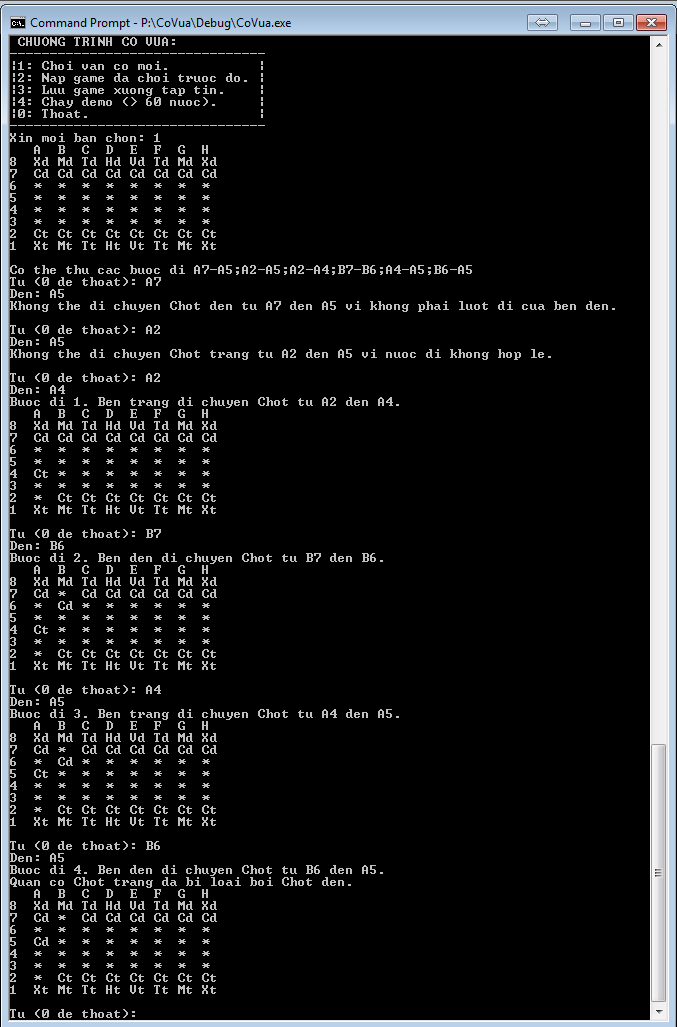
File được đọc từ hàng 0 đến hàng 8. Với tám hàng đầu tiên, các quân cờ được gán vào bàn cờ theo vị trí tương ứng từ hàng 0-7 và cột 0-7. Các cột cách nhau một khoảng trắng. Nếu giá trị tại một cell là \*, NULL được gán cho vị trí tương ứng của bàn cờ. Ngược lại, loại quân cờ, tên quân cờ, màu, trạng thái (sống) được gán chọ vị trí tương ứng của bàn cờ.

Giá trị tại hàng 8 được gán cho số bước đi của bàn cờ. Từ đó ta có thể xác định được bên nào sẽ được đi tiếp theo.

1. **Sử dụng chương trình**

Chương trình cho phép người dùng có thể bắt đầu một ván cờ mới hoặc chơi từ ván cờ cũ. Cơ chế di chuyển của tất cả 6 loại quân cờ đã được cài đặt. Chương trình cho phép lưu game hoặc nạp game đã chơi trước đó.

* **Hình ảnh các nước đi đầu tiên khi chơi một ván cờ mới:**

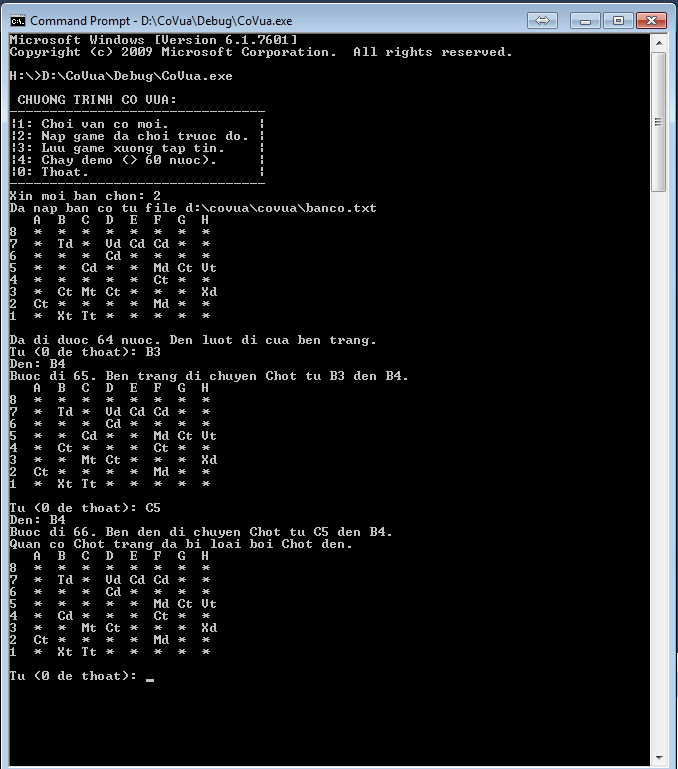


**Tiến hành di chuyển quân cờ đúng hoặc sai luật chơi và kiểm tra vị trí sau khi di chuyển**

* + Bước chuyển quân cờ từ "A7" đến "A5" là không hợp lệ vì bên trắng phải được đi trước.
  + Bước chuyển quân cờ từ "A2" đến "A5" là không hợp lệ vì ở bước xuất phát chốt chỉ đi được từ 1 đến 2 hàng lên phía trước, ở đây đã yêu cầu di chuyển đến 3 hàng.

**Thử cho 1 quân cờ đen ăn quân cờ trắng thì kết quả 2 quân cờ như thế nào?**

* + Chốt trắng bị ăn bởi chốt đen sau 4 nước đi hợp lệ. Trên bàn cờ bây giờ khuyết mất một chốt trắng.
* **Hình ảnh nạp một ván cờ cũ và tiếp tục chơi:**

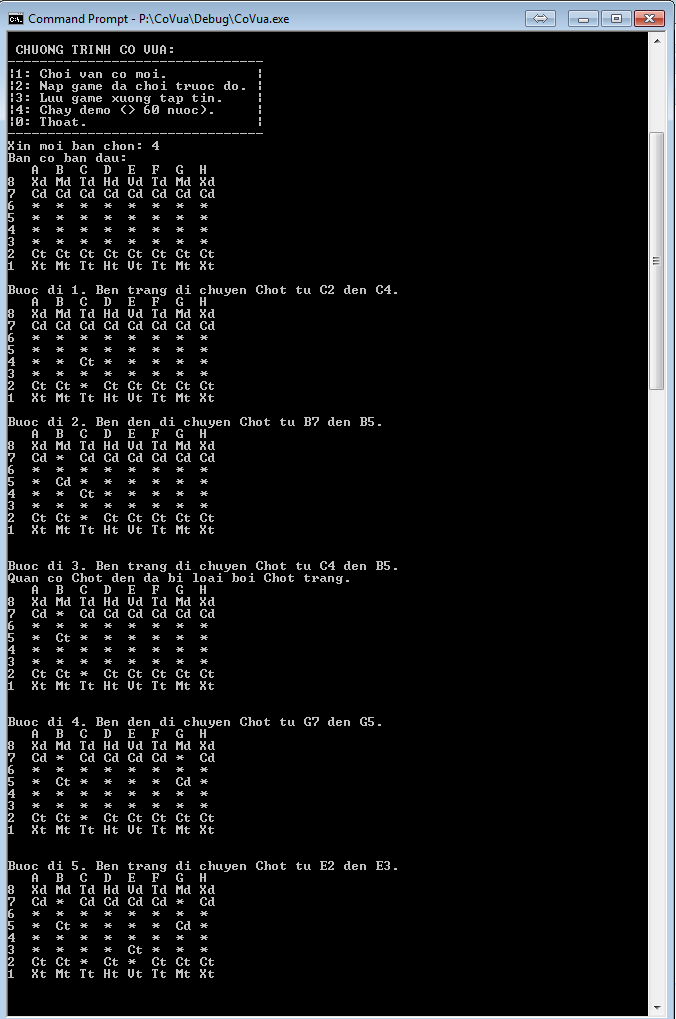


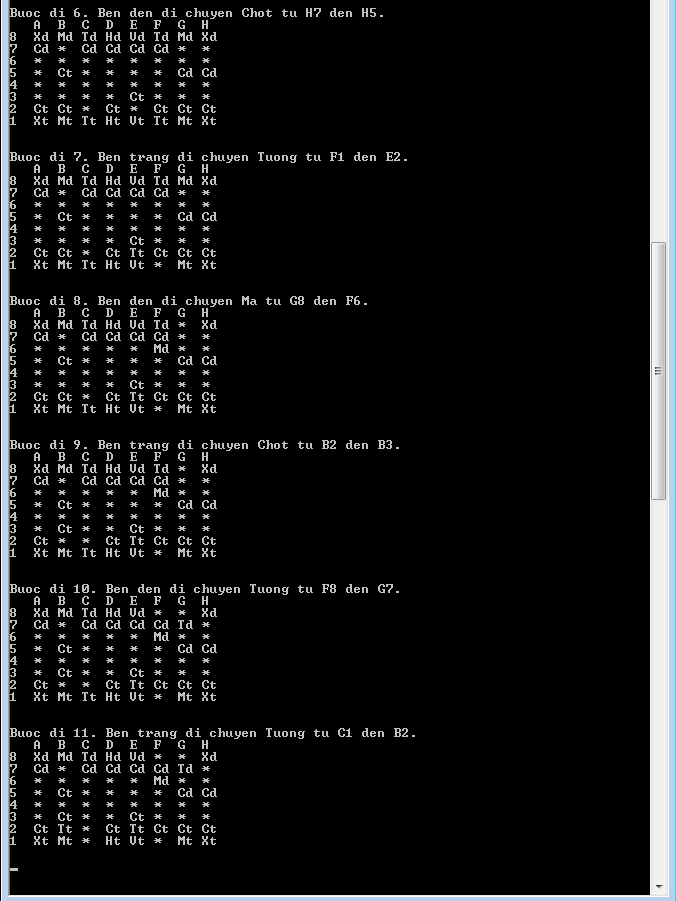
* **Hình ảnh lưu ván cờ đang chơi và kiểm tra bằng cách nạp lại xem có lưu đúng không**

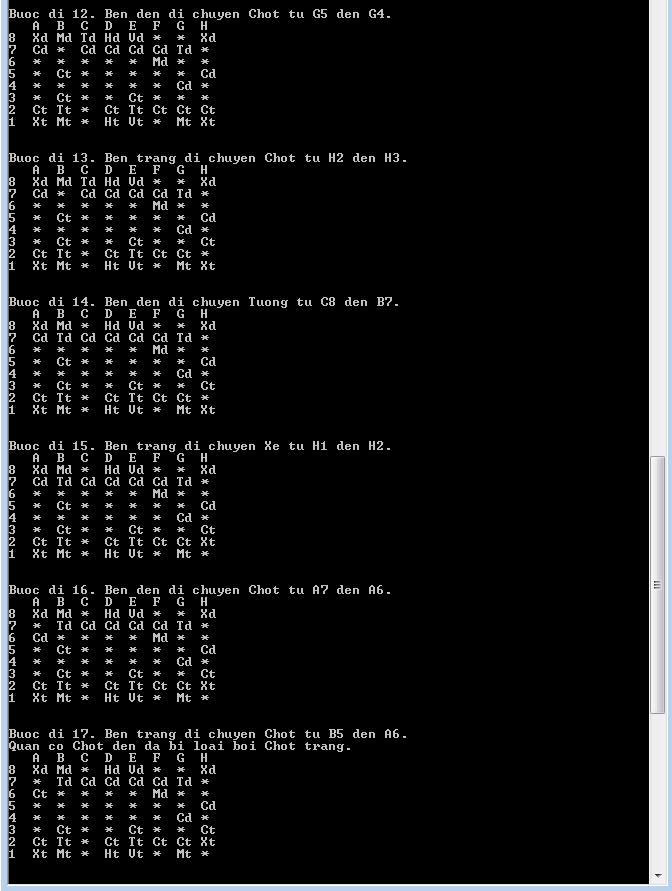


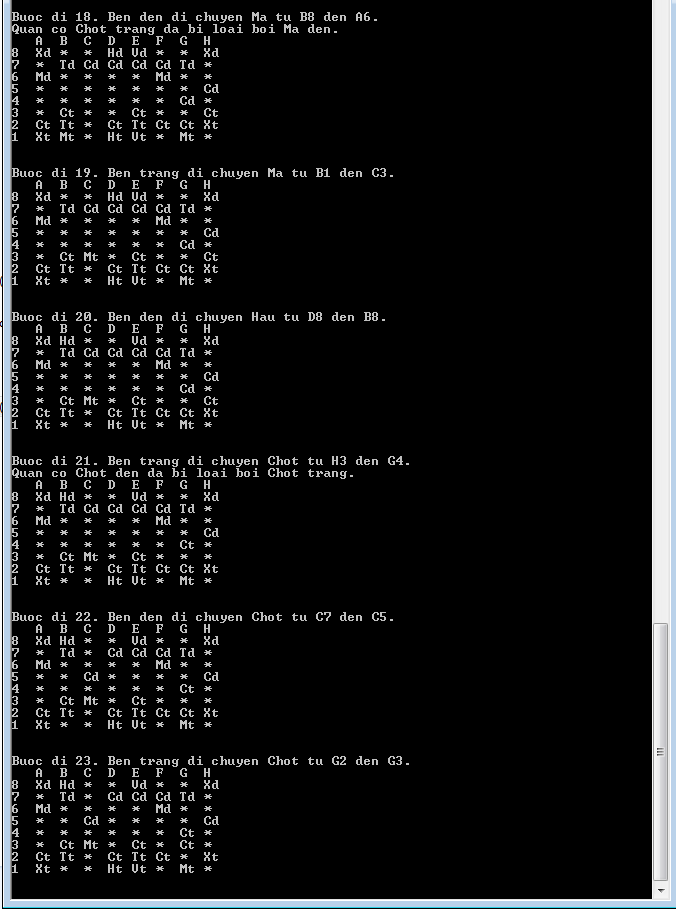
1. **Demo**

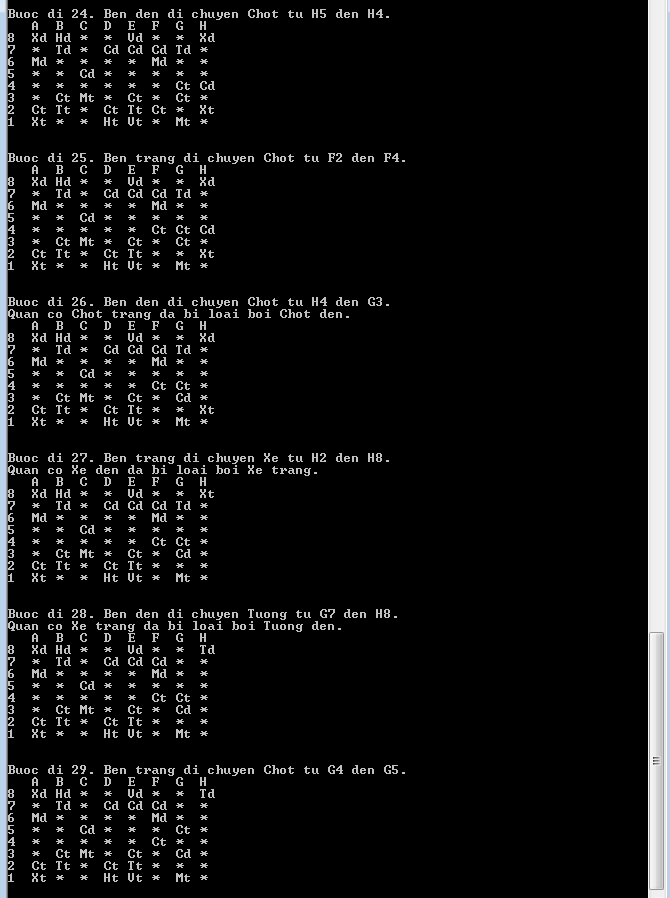
* **Hình ảnh 64 nước đi hợp lệ:**

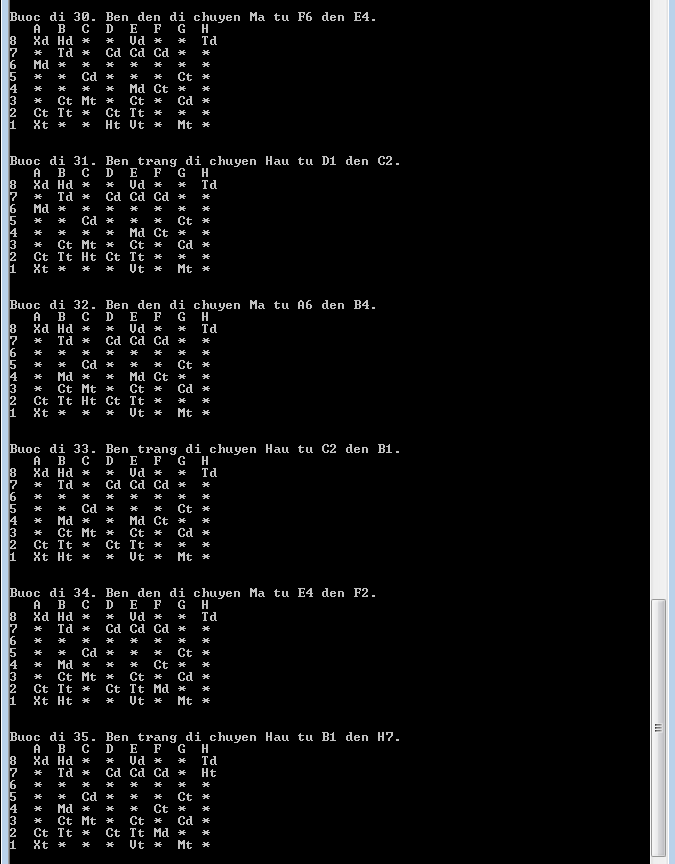
****

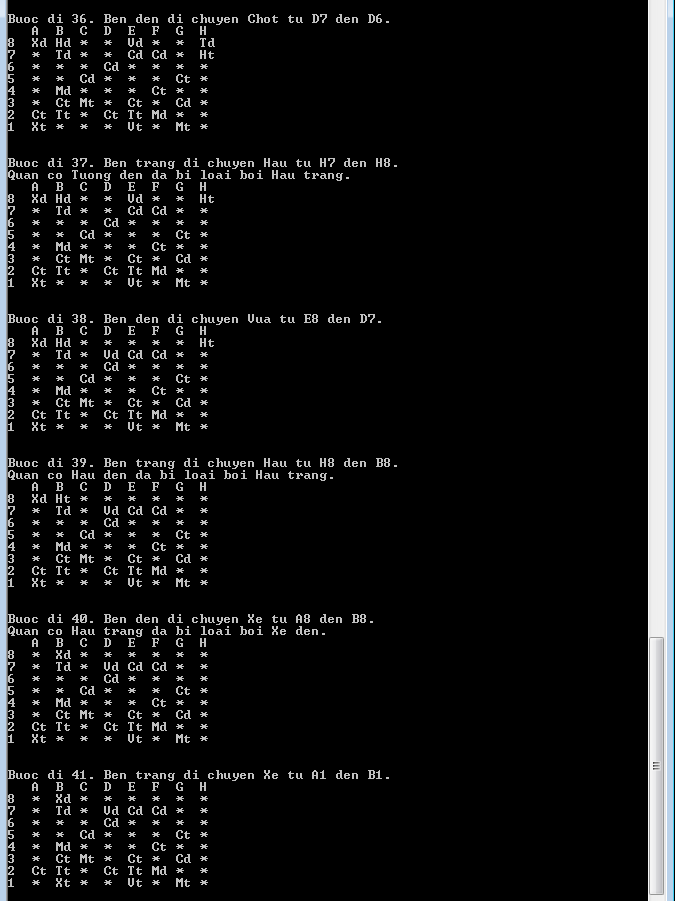


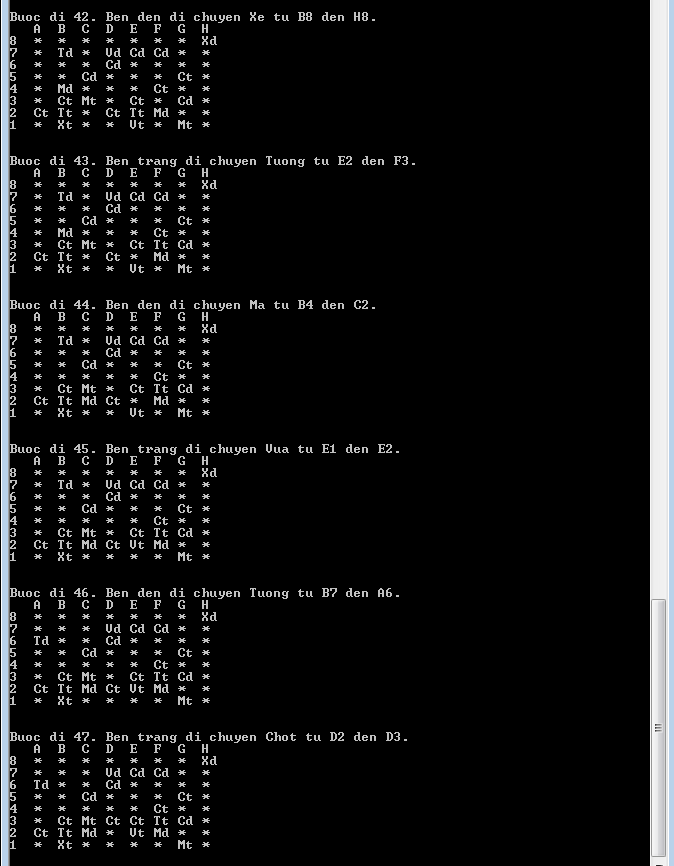


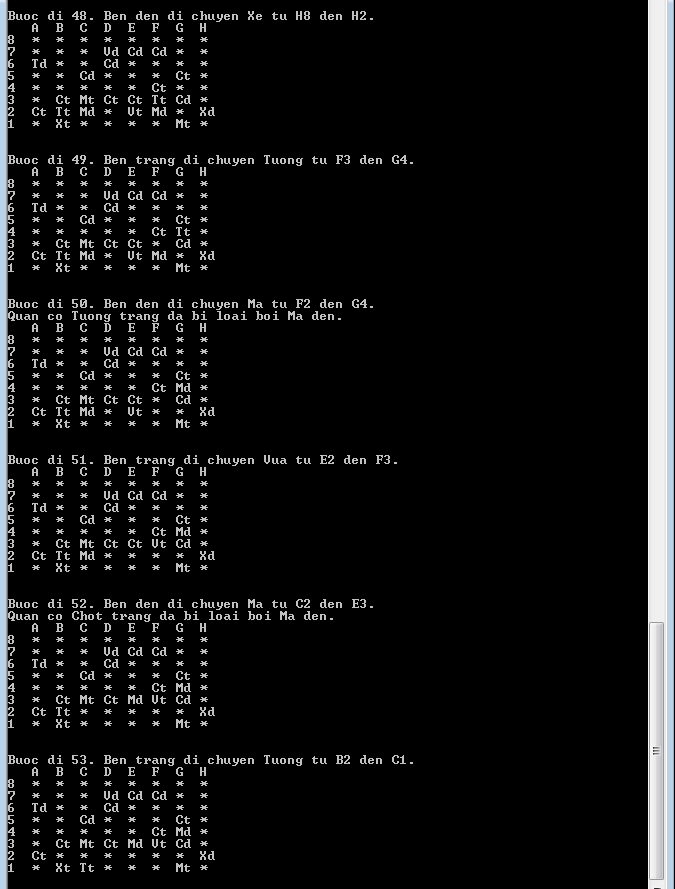


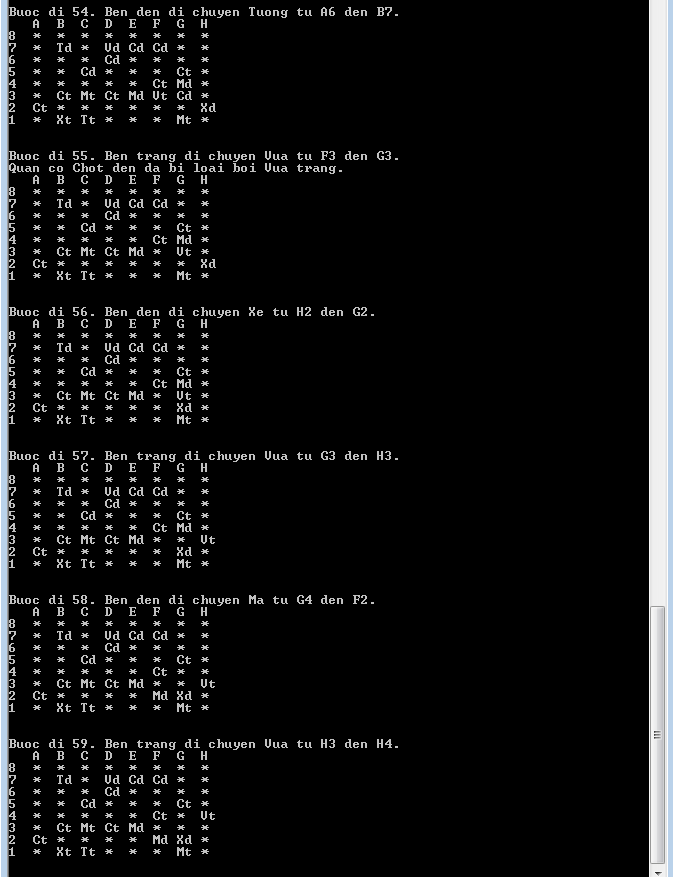


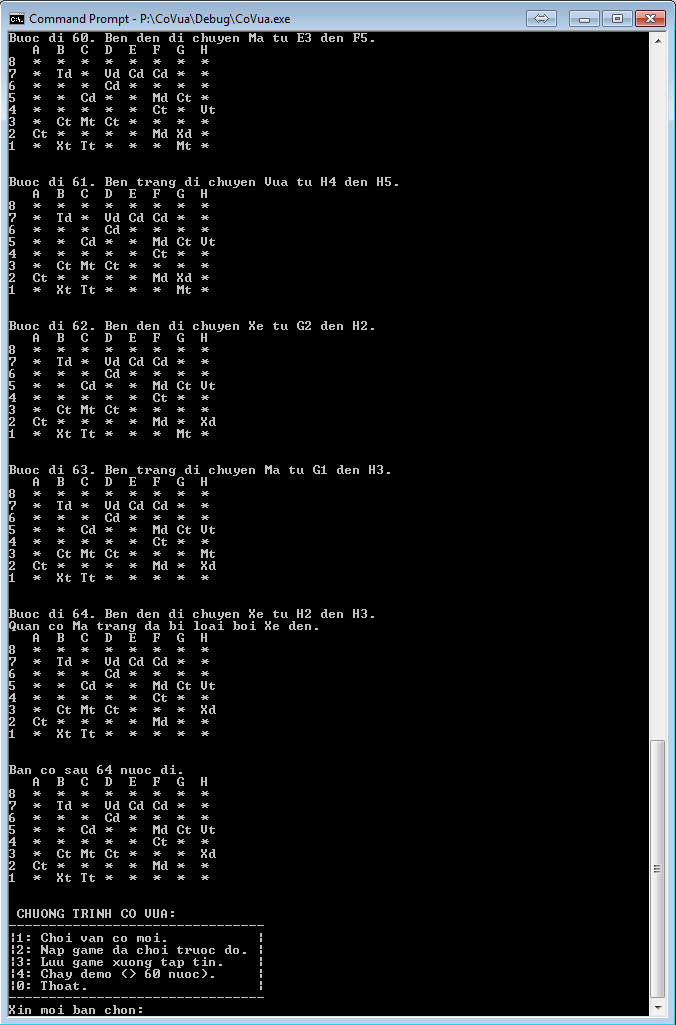












**Câu hỏi dành cho bài tập lớn - Cờ vua**

1. Bạn có áp dụng mẫu thiết kế nào không? Nếu có, hãy chỉ ra và giải thích.

2. Làm sao để thêm một loại quân cờ mới vào trò chơi? Hãy chỉ rõ các bước thực hiện.

3. Nếu như chương trình phải thay đổi cách hiển thị ra màn hình? Những lớp nào của chương trình bạn phải thay đổi? Bạn có thể điều chỉnh thiết kế lớp để giảm sự thay đổi này không?

4. Đối với tình huống Tốt được phong thành quân nặng (Hậu, Xe, Tượng, Mã), chương trình của bạn xử lý như thế nào?

Trả lời:

1. Có thể xem lớp Quân Cờ như kiểu thiết kế A**bstract** Factory. Khi đó các lớp đều có chung các thuộc tính, chỉ khác nhau cách vận hành. Các lớp cụ thể như Vua, Hậu… là các Concrete Factory của lớp Quân Cờ.
2. Khi thêm một loại quân cờ mới vào trò chơi, ta chỉ cần cho lớp đó thừa kế tất cả các thuộc tính của lớp Quân Cờ và cài đặt bước đi thích hợp cho loại quân cờ đó. Chẳng hạn, với quân Mã, ta cài đặt như sau:

class Ma: public QuanCo{

public:

bool move(int hangmoi, int cotmoi, QuanCo\* \*\*banco)

{

// Khong di chuyen duoc neu quan co o dich den cung mau

if((banco[hangmoi][cotmoi] != NULL) && (banco[hangmoi][cotmoi] -> trang == trang))

return false;

if(((hangmoi - hang == 2) && (cotmoi - cot == 1)) ||

((hangmoi - hang == 2) && (cotmoi - cot == -1)) ||

((hangmoi - hang == 1) && (cotmoi - cot == 2)) ||

((hangmoi - hang == 1) && (cotmoi - cot == -2)) ||

((hangmoi - hang == -1) && (cotmoi - cot == 2)) ||

((hangmoi - hang == -1) && (cotmoi - cot == -2)) ||

((hangmoi - hang == -2) && (cotmoi - cot == 1)) ||

((hangmoi - hang == -2) && (cotmoi - cot == -1)))

{

return true;

}

else

return false;

return false;

}

};

Khi đó quân Mã sẽ thừa kế các thuộc tính sau của lớp Quân Cờ:

public:

int hang;// quan co nam o hang nao?

int cot; // quan co nam o cot nao?

bool trang;// co phai mau trang hay khong?

bool song;//con song khong?

std::string ten;//ten quan co

Và có bước đi hợp lệ như cài đặt trên.

1. Chương trình đã được thiết kế tinh giản để khi hiển thị ra màn hình, ta chỉ cần chỉnh sửa lớp Bàn Cờ. Ở bước khởi tạo cũng như sau mỗi bước di chuyển, ta chỉ cần kiểm tra tại mỗi ô có quân cờ hay không, nếu có thì đó là quân gì, màu gì, và hiển thị lên bàn cờ.

Sau khi xử lý như bình thường cho một bước di chuyển quân cờ, ta thêm đoạn mã xử lý đối với quân Chốt như sau:

if(banco[den\_hang][den\_cot]->ten.compare("Chot")==0)

{

while(true)

{

int quan;

cout << "Ban muon Chot thanh quan nao? 0: Chot, 1: Hau, 2: Xe, 3: Tuong, 4: Ma"; cin >> quan;

bool contains\_non\_04 = !std::regex\_match(tu.substr(1), std::regex("^[0-4]+$"));

if(contains\_non\_04)

cout << "Nhap sai, xin moi nhap lai."<<endl;

else

{

if(quan==0)

break;// Giu nguyen Chot, khong phong thanh quan nang.

else

{

string mau = "d";

if(banco[den\_hang][den\_cot]->trang)

mau="t";

string cell="Hau";

if(quan==2)

cell="Xe";

else if(quan==3)

cell="Tuong";

else if(quan==4)

cell="Ma";

cell = cell + mau;

GanQuanCo(cell,den\_hang,den\_cot);

break;

}

}

}

}