

BỘ GIÁO DỤC & ĐÀO TẠO
TRƯỜNG ĐẠI HỌC KINH TẾ TP. HCM
KHOA CÔNG NGHỆ THÔNG TIN KINH DOANH



TIỂU LUẬN CUỐI KÌ
Môn: CƠ SỞ LẬP TRÌNH

Đề bài: Viết trò chơi cá ngựa cho 4 người.

Giảng viên hướng dẫn: ThS. Trần Lê Phúc Thịnh
Lớp học phần: 23C1INF50900502
Lớp: DS001

TP. Hồ Chí Minh, ngày 3, tháng 10, năm 2023

LỜI CẢM ƠN

Để có thể hoàn thành được bài thi cuối kì và môn Cơ sở lập trình, nhóm 1 chúng em xin gửi lời cảm ơn đến thầy Trần Lê Phúc Thịnh, nhờ sự nhiệt tình và tận tâm của thầy trong quá trình giảng dạy đã giúp chúng em có thêm nhiều kiến thức hay, những kỹ năng cần thiết và bổ ích. Thầy đã luôn quan tâm, chỉ bảo, đưa ra những kinh nghiệm cũng như những lời động viên để chúng em làm tốt hơn.

Dù đã rất cố gắng nhưng trong quá trình nghiên cứu, nhưng chúng em vẫn gặp khá nhiều khó khăn nên không thể tránh khỏi các sai sót để hoàn thành bài thi một cách tốt nhất. Vì vậy, chúng em rất mong nhận được những ý kiến đóng góp của thầy để bài làm của chúng em được hoàn thiện hơn.

Nhóm 1 xin chân thành cảm ơn thầy! Chúc thầy nhiều sức khỏe, luôn thành công trong việc giảng dạy và gặp nhiều may mắn trong cuộc sống.

Danh sách thành viên nhóm 1:

31221026992	Võ Yến Nhi (NT)
31221025662	Nguyễn Thành Vinh
31221026291	Bùi Tiến Hiếu
31221024364	Phạm Bằng
31221026655	Nguyễn Trần Thế Anh
33231020124	Nguyễn Thùy Yến Nhi
31221025556	Đặng Thị Thu Hiền
31221024212	Trương Phạm Bảo Khanh

Mục lục

I. Giới thiệu C#.....	1
II. Giới thiệu Cờ Cá Ngựa.....	1
1. Tổng quan về Cờ Cá Ngựa.....	1
2. Luật chơi và cách chơi Cờ Cá Ngựa.....	1
2.1. Luật chơi	1
2.2. Cách chơi.....	2
II. Cấu trúc dữ liệu sử dụng trong bài:.....	3
1. Xử lý đối với phần “ô đường đi”:	3
2. Xử lý đối với phần “ô về đích”:	3
3. Cách thức hoạt động:	4
III. Sơ đồ khối.....	4
1. Hàm oco (VẼ BÀN CỜ).....	4
2. Hàm cellfinish (VẼ BÀN CỜ):	5
3. Vẽ chuồng ngựa (VẼ BÀN CỜ)	5
4. Thiết lập toạ độ để tạo thành ô đường đi và ô về đích (VẼ BÀN CỜ):	7
5. Vẽ con cờ trong chuồng (VẼ BÀN CỜ):	8
6. Hàm ve_chuong(mau_co,stt).....	9
7. Hàm mau(mau)	10
8. Hàm xx().....	11
9. Hàm back_home(x)	11
10. Hàm check_go(y,x)	12
11. Hàm check_finish(y,x)	13

12. Hàm gogo(x,stt)	13
13. Hàm auto()	14
14.Hàm quanco.....	15
15. Hàm Main	16
IV. Chạy demo và code	17
1. Chạy demo	17
2. Code	23

I. Giới thiệu C#

– C# (hay C Sharp) là một ngôn ngữ lập trình thuần hướng đối tượng, được phát triển bởi Microsoft vào năm 2000. Nó là một phần của nền tảng phát triển phần mềm .NET Framework của Microsoft và được sử dụng chủ yếu để xây dựng các ứng dụng Windows và ứng dụng web, được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java.

– Các phiên bản C#: C# 1.0 (2002), C# 5.0 (2010), C# 7.0, 8.0 và 9.0 (từ năm 2017 đến năm 2020),...

– C# được sử dụng rộng rãi trong việc phát triển các ứng dụng Windows, bao gồm ứng dụng desktop, ứng dụng di động phát triển game, phát triển web và máy tính đám mây. Ngoài ra, C# cũng là ngôn ngữ được sử dụng chính trong việc phát triển các ứng dụng web với ASP.NET và ASP.NET Core.

II. Giới thiệu Cờ Cá Ngựa

1. Tổng quan về Cờ Cá Ngựa

- Trò chơi Cờ Cá Ngựa có nguồn gốc từ thế kỷ 16 và được phát triển từ trò chơi cờ cung hoàng đạo của Ấn Độ. Trò chơi yêu cầu từ 2 đến 4 người chơi và mỗi người chơi sẽ điều khiển một đội ngựa gồm 4 quân cờ trên bàn cờ hình vuông.

- Một bộ cờ cá ngựa bao gồm:

- 1 bàn cờ hình vuông được chia làm 4 khu vực có một màu đại diện (đỏ, vàng, xanh dương, xanh lá).
- 16 quân cờ được chia ra 4 màu tương ứng với màu của 4 khu vực.
- 1 viên xúc xắc.

- Mục tiêu của trò chơi là đưa tất cả các quân cờ từ khung xuất phát đến khung kết thúc. Để xác định số nước đi cho các quân ngựa của mình, bạn sẽ phải thực hiện việc tung xúc xắc. Trên đường đi, bạn sẽ phải tránh các quân cờ của đối thủ và cố gắng sử dụng các ô an toàn để tránh bị đánh bại.

- Trò chơi Cờ Cá Ngựa không chỉ đòi hỏi sự may mắn từ việc tung xúc xắc mà còn yêu cầu người chơi phải có chiến thuật và khả năng quyết đoán. Bạn phải xác định đường đi tốt nhất cho các quân cờ của mình và cân nhắc kỹ lưỡng để tránh bị đối thủ chặn đường hoặc đánh bại.

2. Luật chơi và cách chơi Cờ Cá Ngựa

2.1. Luật chơi

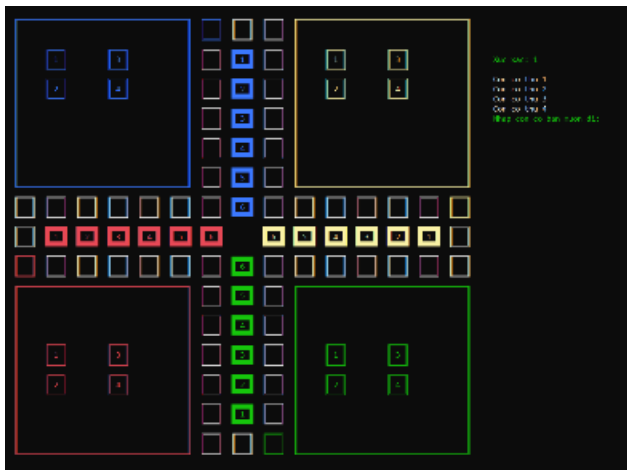
- Cờ cá ngựa có tối đa 4 người chơi. Mỗi người chơi sẽ có 1 chuồng và 4 quân cờ trong chuồng.

- Những người chơi sẽ lần lượt tung xúc xắc và di chuyển các quân cờ của mình từ ô bắt đầu tới khi đủ 1 vòng bàn cờ (ngược chiều kim đồng hồ).
- Sau khi di chuyển hết 1 vòng bàn cờ thì người chơi sẽ phải đưa các quân cờ của mình về các ô chuồng được in đậm và đánh số 6, 5, 4, 3.

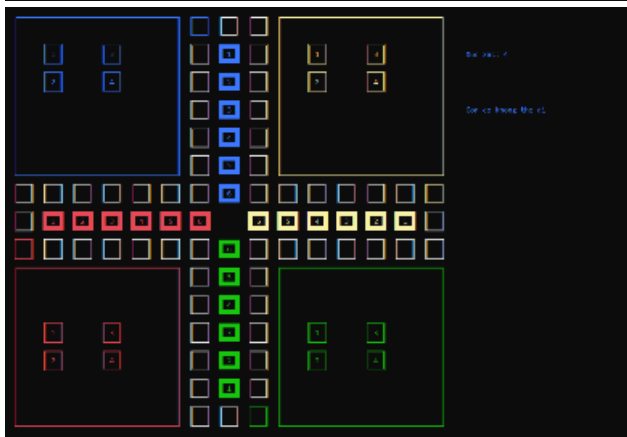
- Người chơi nào có 4 con ngựa về các ô chuồng như trên đầu tiên sẽ là người chiến thắng.

2.2. Cách chơi

- Những người chơi sẽ lần lượt tung xúc xắc, theo thứ tự xanh dương đầu tiên, sau đó lần lượt là đỏ, xanh lá, vàng và thứ tự sẽ tiếp tục được lặp lại như vậy đến hết ván cờ.



TH1: Nếu như tung xúc xắc trúng ‘1’ hoặc ‘6’ thì người chơi sẽ được chọn 1 trong 4 quân cờ của mình để xuất chuồng hoặc di chuyển các quân cờ đã xuất chuồng. Trường hợp này người chơi đó tiếp tục có thêm lượt tung xúc xắc mới.



TH2: Nếu xúc xắc không trúng ‘1’ hoặc ‘6’ thì người chơi sẽ chỉ được chọn một trong những quân cờ đã xuất chuồng để di chuyển. Nếu như chưa có quân cờ nào xuất chuồng thì người chơi sẽ được thông báo “Quân cờ không thể đi” và lượt chơi sẽ được chuyển cho người tiếp theo.

* Nếu như có một quân cờ (cùng hoặc khác màu) nằm trên đường đi của một quân cờ cùng màu khác thì quân cờ cùng màu khác đó sẽ không thể di chuyển.

* Trường hợp có một quân cờ có màu khác nằm ở vị trí mà một quân cờ đi tới thì quân cờ có màu khác sẽ bị đá quay về chuồng.

- Trò chơi sẽ lần lượt được tiếp tục theo vòng tròn ngược chiều kim đồng hồ như vậy cho tới khi các quân cờ đi hết 1 vòng tròn. Để quân cờ có thể leo lên được “các ô về đích”, quân cờ đó phải đứng ở vị trí cuối cùng (ngay dưới “đích”) của nó, sau đó, khi xúc xắc quay ra nút bao nhiêu thì con cờ đó sẽ leo lên bấy nhiêu “ô về đích” nếu trên đường đi của nó không tồn tại con cờ cản đường. Lúc đó, người chơi nào đưa các quân cờ về các ô chuồng được tô đậm và đánh số 6, 5, 4, 3 đầu tiên sẽ là người chiến thắng.

II. Cấu trúc dữ liệu sử dụng trong bài:

- Trong bàn cờ có 4 màu cờ với các quy ước sau:
 - o Xanh dương – 0
 - o Đỏ - 1
 - o Xanh lá – 2
 - o Vàng - 3
- Bàn cờ được chia thành 3 phần, tạm gọi là: chuồng, ô đường đi, ô về đích.

1. Xử lý đối với phần “ô đường đi”:

+ Có 56 ô đường đi trên bàn cờ và được đánh số thứ tự (từ số 0) ngược chiều kim đồng hồ, bắt đầu từ “ô ra chuồng” của con cờ xanh dương.

+ Như vậy, “ô ra chuồng” của quân cờ màu xanh dương, đỏ, xanh lá, vàng lần lượt là 0, 14, 28, 42.

+ Mảng “vitri” chứa các giá trị tổng quát: “ a^b ” với ‘a’ là cột, ‘b’ là hàng. Mỗi phần tử trong mảng **vitri[x]** ứng với tọa độ của một ô cờ theo đúng thứ tự ngược chiều kim đồng hồ như trên.

```
static public string[] vitri =
```

```
{
```

```
    "6^0", "6^1", "6^2", "6^3", "6^4", "6^5", "6^6", "5^6", "4^6", "3^6", "2^6",  
    "1^6", "0^6", "0^7", "0^8", "1^8", "2^8", "3^8", "4^8", "5^8", "6^8", "6^9", "6^10", "6^11", "6^12",  
    "6^13", "6^14", "7^14", "8^14", "8^13", "8^12", "8^11", "8^10", "8^9", "8^8", "9^8", "10^8", "11^8",  
    "12^8", "13^8", "14^8", "14^7", "14^6", "13^6", "12^6", "11^6", "10^6", "9^6", "8^6", "8^5", "8^4",  
    "8^3", "8^2", "8^1", "8^0", "7^0"
```

```
};
```

+ Mảng “**tontai[i]**”: mỗi phần tử **tontai[i]** ($i = 0, 1, \dots, 55$) trong mảng lưu 1 trong 2 giá trị sau đây:

- **0**: tức là tại ô thứ i không tồn tại con cờ.
- **\overline{ab}** : tại ô thứ i tồn tại con cờ có màu là ‘a’, số thứ tự là ‘b’.

+ Mảng “**kt[i, j]**” là mảng hai chiều, **kt[i, j]** nhận 1 trong 2 giá trị:

- **false**: con cờ có màu ‘i’, số thứ tự j chưa ra chuồng
- **true**: con cờ có màu ‘i’, số thứ tự j đã ra chuồng.

+ Mảng “**buoc**” là mảng 2 chiều, **buoc[i, j]** ý nghĩa là con cờ có màu thứ ‘i’, số thứ tự j đã đi được **buoc[i, j]** bước.

2. Xử lý đối với phần “ô về đích”:

+ Tương tự đối với phần xử lý “ô đường đi”, phần xử lý “ô về đích” cũng có các mảng như:

- **lentroi[x]**: có chức năng giống mảng **vitri[x]**, nhưng nó lưu tọa độ các ô về đích.

- **tontai[i,x]**: có chức năng giống mảng **tontai[x]**.
- **finish[i,j]**: có chức năng giống mảng **buoc[i,j]**.

3. Cách thức hoạt động:

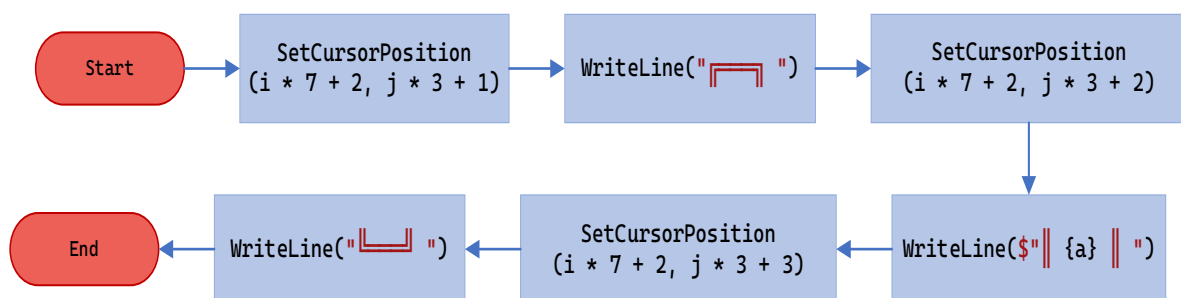
Khi đến lượt người chơi có con cờ màu ‘A’, người chơi sẽ chọn con cờ thứ ‘B’ ($1 \leq B \leq 4$). Từ hai biến này, ta suy ra được số bước đi của con cờ đó (tạm gọi là con cờ ‘AB’). Có được số bước của con cờ ‘AB’, chúng ta rút ra được vị trí hiện tại của nó. Như vậy, các hoạt động sau đó dễ dàng được tính theo.

III. Sơ đồ khối

1. Hàm ọco (VẼ BÀN CỜ)

- Dùng để vẽ ô cờ đơn vị tại cột thứ i, hàng thứ j và bên trong chứa chuỗi ‘a’.

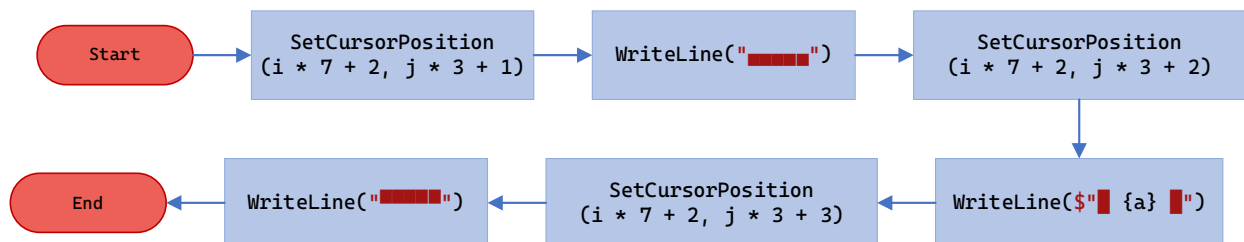
```
static void ọco(string a, int i, int j)
{
    Console.SetCursorPosition(i * 7 + 2, j * 3 + 1);
    Console.WriteLine("┌───┐");
    Console.SetCursorPosition(i * 7 + 2, j * 3 + 2);
    Console.WriteLine($"│ {a} │");
    Console.SetCursorPosition(i * 7 + 2, j * 3 + 3);
    Console.WriteLine("└───┘");
}
```



2. Hàm cellfinish (VẼ BÀN CỜ):

- Dùng để vẽ ô cờ về đích tại cột thứ i, hàng thứ j và bên trong chứa chuỗi "a".

```
static void cellfinish(string a, int i, int j)
{
    Console.SetCursorPosition(i * 7 + 2, j * 3 + 1);
    Console.WriteLine("■");
    Console.SetCursorPosition(i * 7 + 2, j * 3 + 2);
    Console.WriteLine($"■ {a} ■");
    Console.SetCursorPosition(i * 7 + 2, j * 3 + 3);
    Console.WriteLine("■");
}
```



3. Vẽ chuồng ngựa (VẼ BÀN CỜ)

Dưới đây là ví dụ về màu xanh dương, các màu khác tương tự.

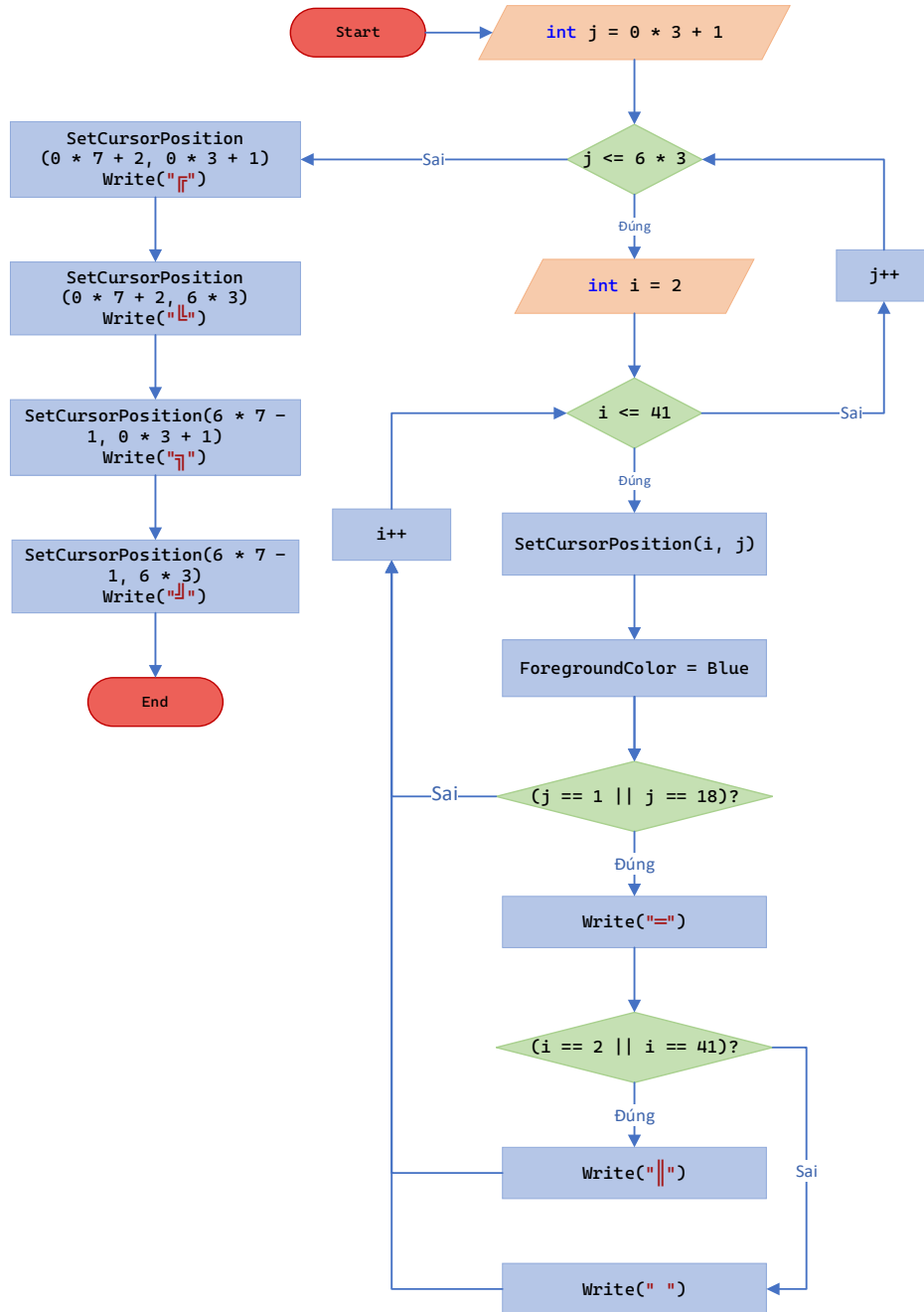
// Tạo ra một khung chữ nhật

Tạo ra một hình chữ nhật có viền màu xanh dương, với các vị trí bên trong được điền bằng khoảng trắng.

```
for (int j = 0 * 3 + 1; j <= 6 * 3; j++)
{
    for (int i = 2; i <= 41; i++)
    {
        Console.SetCursorPosition(i, j);
        Console.ForegroundColor = ConsoleColor.Blue;
        if (j == 1 || j == 18) Console.Write("=");
        else if (i == 2 || i == 41) Console.Write("||");
        else Console.Write(" ");
    }
}
```

// Vẽ 4 góc

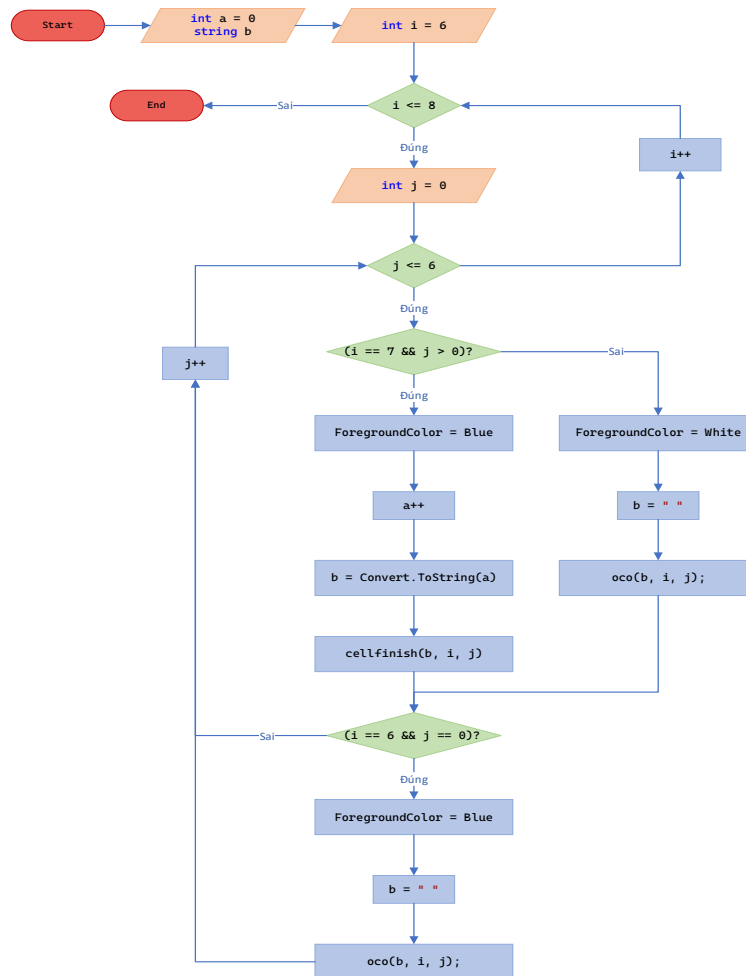
```
Console.SetCursorPosition(0 * 7 + 2, 0 * 3 + 1); Console.Write("┌");
Console.SetCursorPosition(0 * 7 + 2, 6 * 3); Console.Write("└");
Console.SetCursorPosition(6 * 7 - 1, 0 * 3 + 1); Console.Write("┐");
Console.SetCursorPosition(6 * 7 - 1, 6 * 3); Console.Write("┘");
```



4. Thiết lập tọa độ để tạo thành ô đường đi và ô về đích (VỀ BÀN CỜ):

Dưới đây là ví dụ về màu xanh dương, các màu khác tương tự

```
int a = 0; string b;
for (int i = 6; i <= 8; i++)
    for (int j = 0; j <= 6; j++)
    {
        // Xét các ô về đích
        if (i == 7 && j > 0) // Tô màu xanh dương
        {
            Console.ForegroundColor = ConsoleColor.Blue;
            a++;
            b = Convert.ToString(a);
            cellfinish(b, i, j);
        }
        else // Tô màu trắng
        {
            Console.ForegroundColor = ConsoleColor.White;
            b = " ";
            oco(b, i, j);
        }
        if (i == 6 && j == 0) // Xét ô ra quân
        {
            Console.ForegroundColor = ConsoleColor.Blue; // Tô màu xanh dương
            b = " ";
            oco(b, i, j);
        }
    }
}
```



5. Vẽ con cờ trong chuỗi (VẼ BÀN CỜ):

Dưới đây là ví dụ về màu xanh dương, các màu khác tương tự

// Tô màu xanh dương

```
Console.ForegroundColor = ConsoleColor.Blue;
```

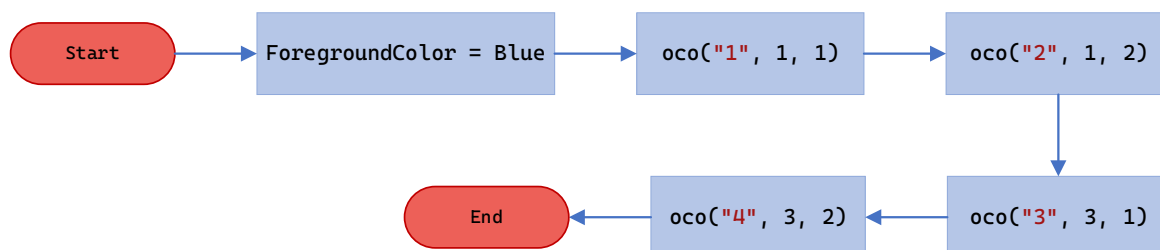
// Vẽ con cờ trong chuỗi

```
oco("1", 1, 1);
```

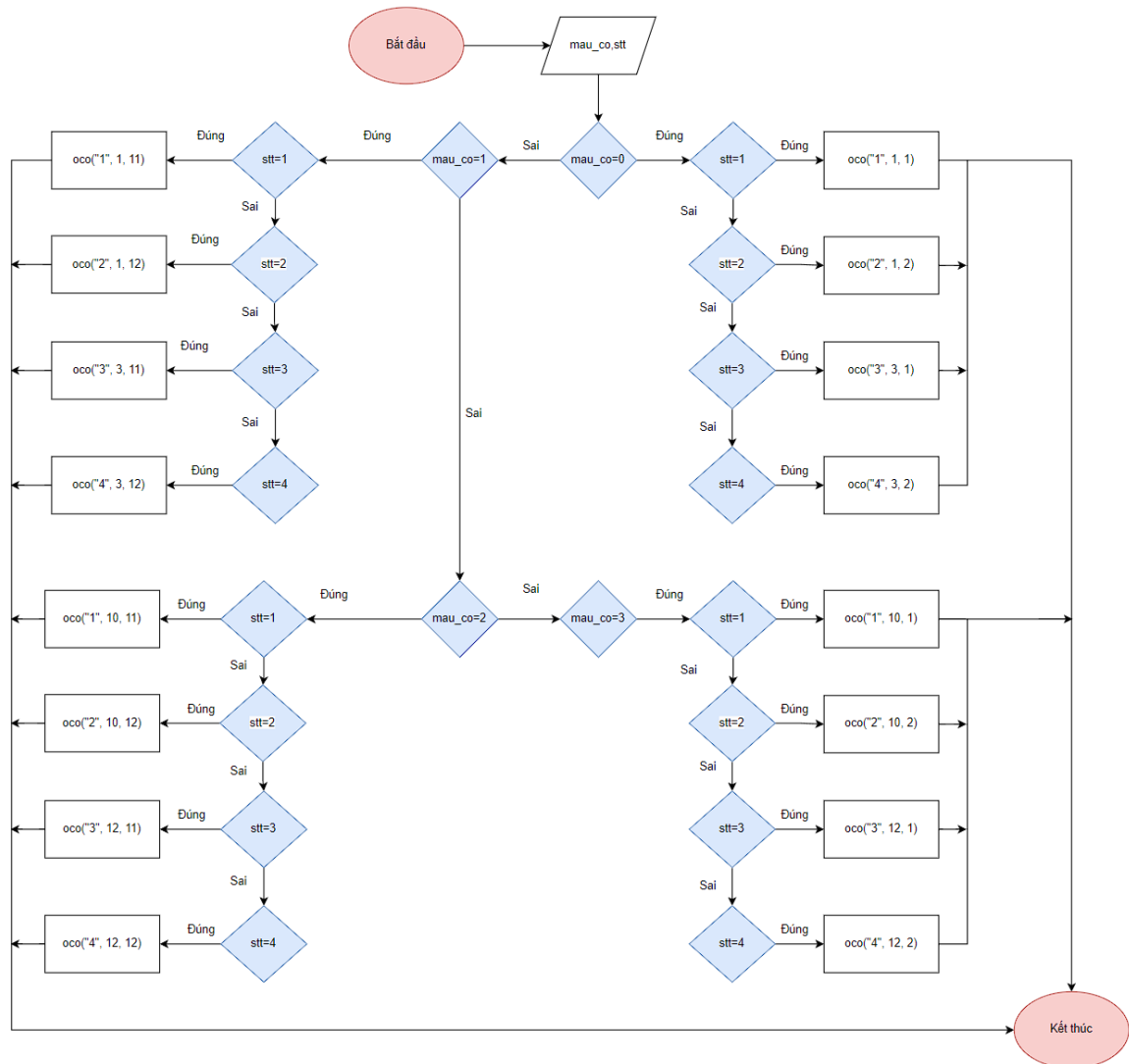
```
oco("2", 1, 2);
```

```
oco("3", 3, 1);
```

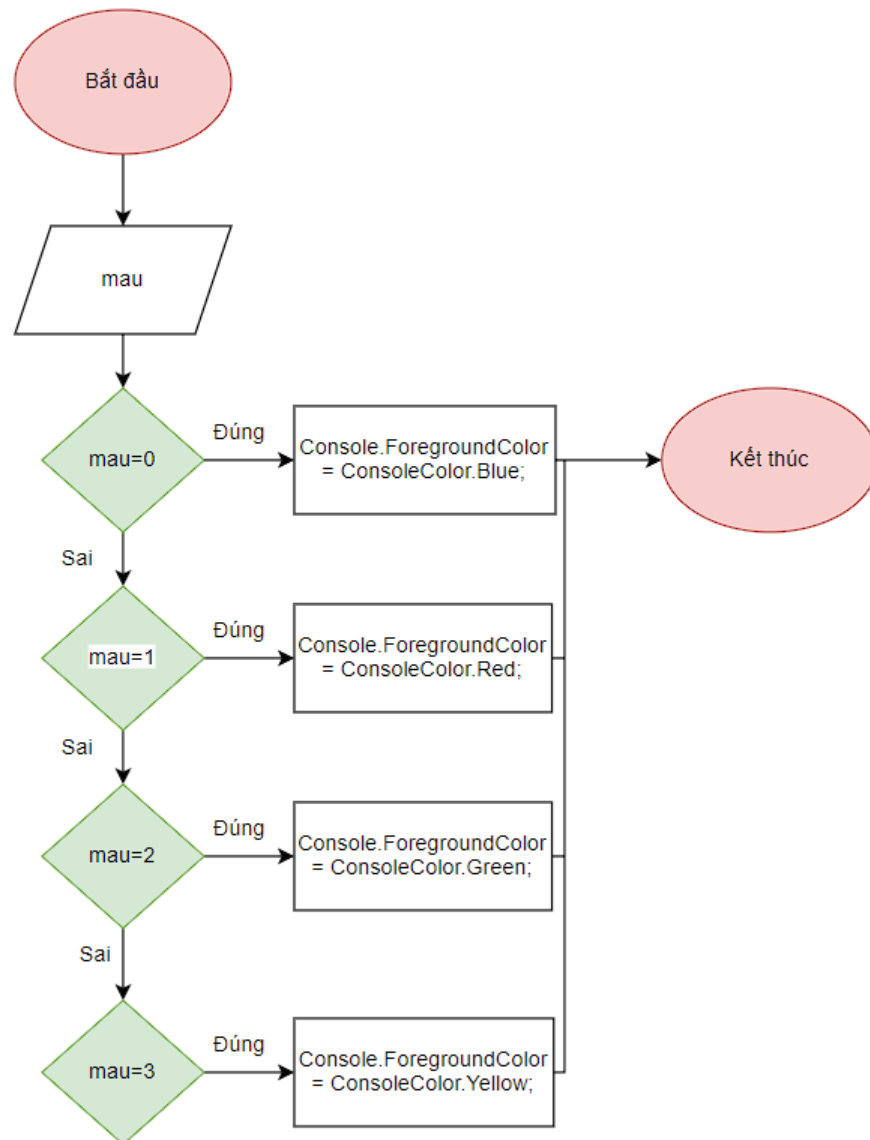
```
oco("4", 3, 2);
```



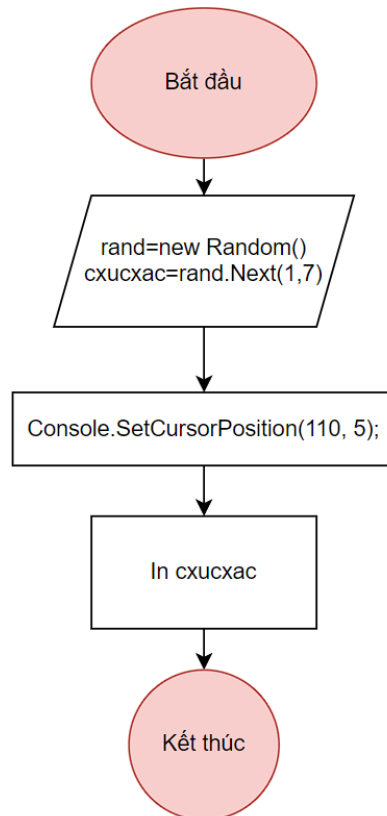
6. Hàm ve_chuong(mau_co,stt)



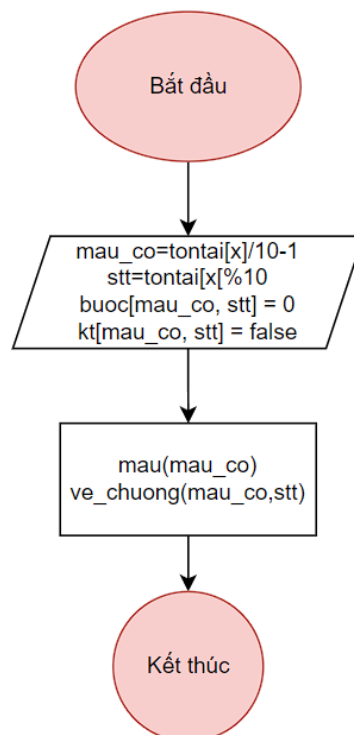
7. Hàm mau(mau)



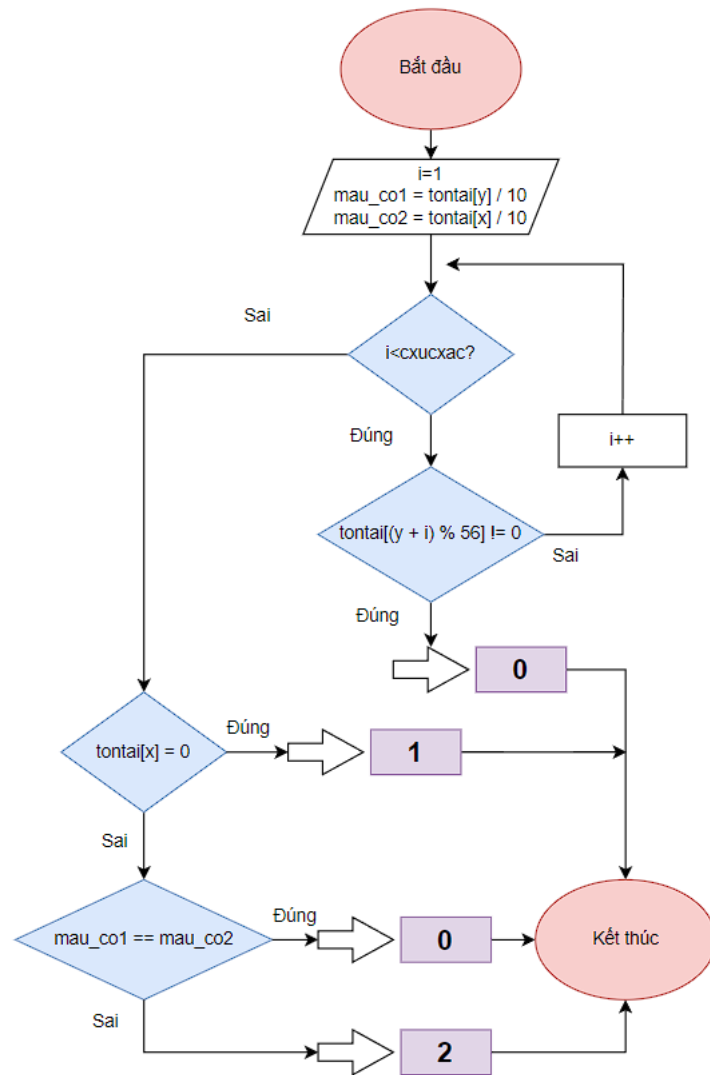
8. Hàm xx()



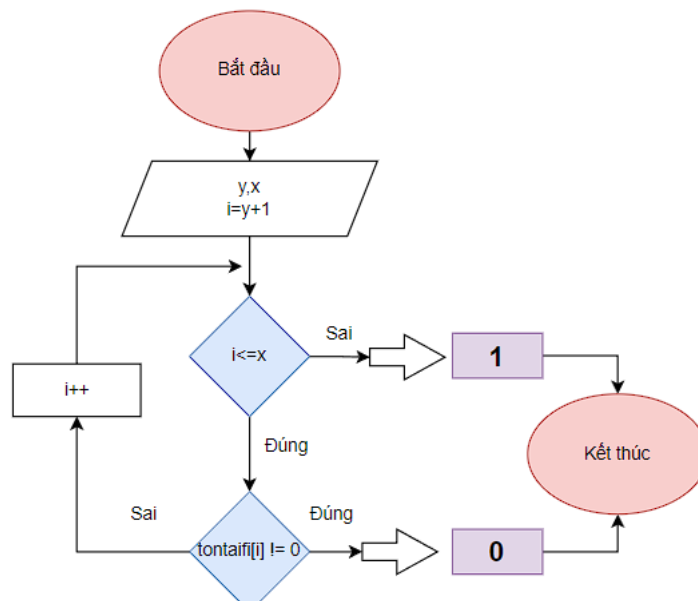
9. Hàm back_home(x)



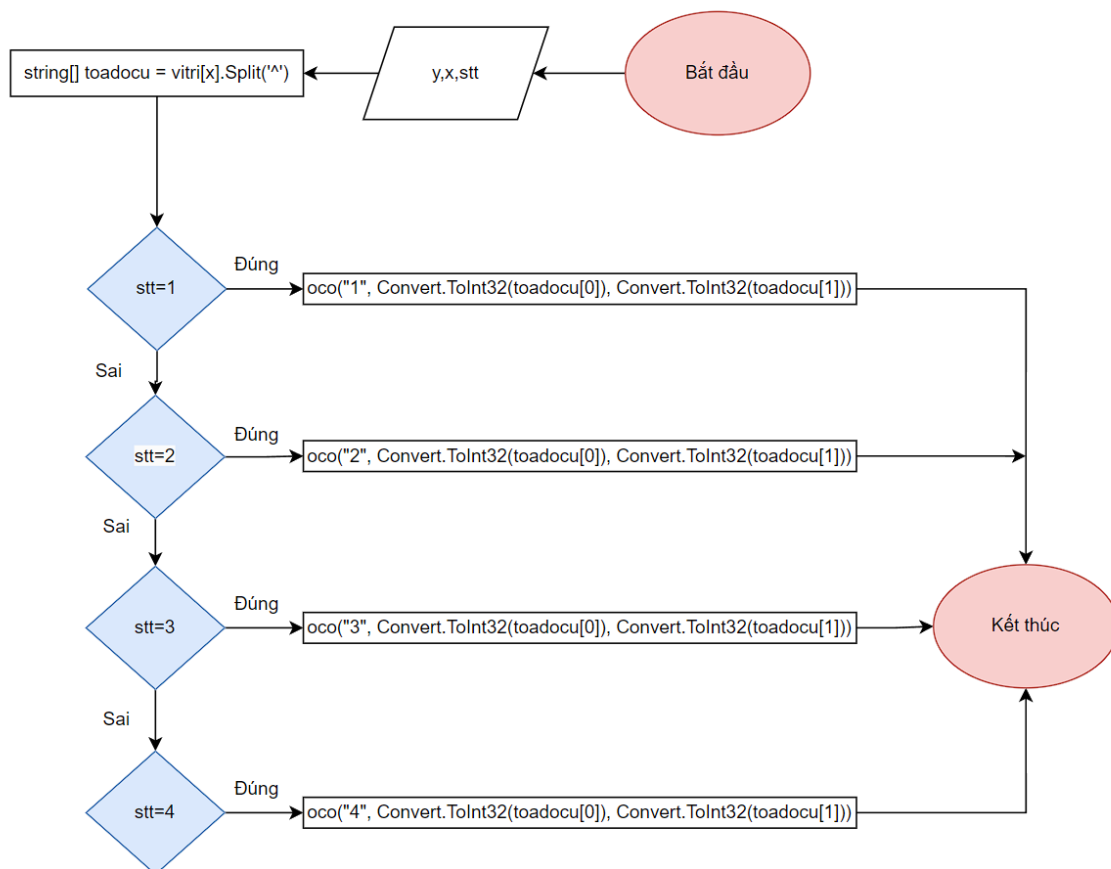
10. Hàm check_go(y,x)



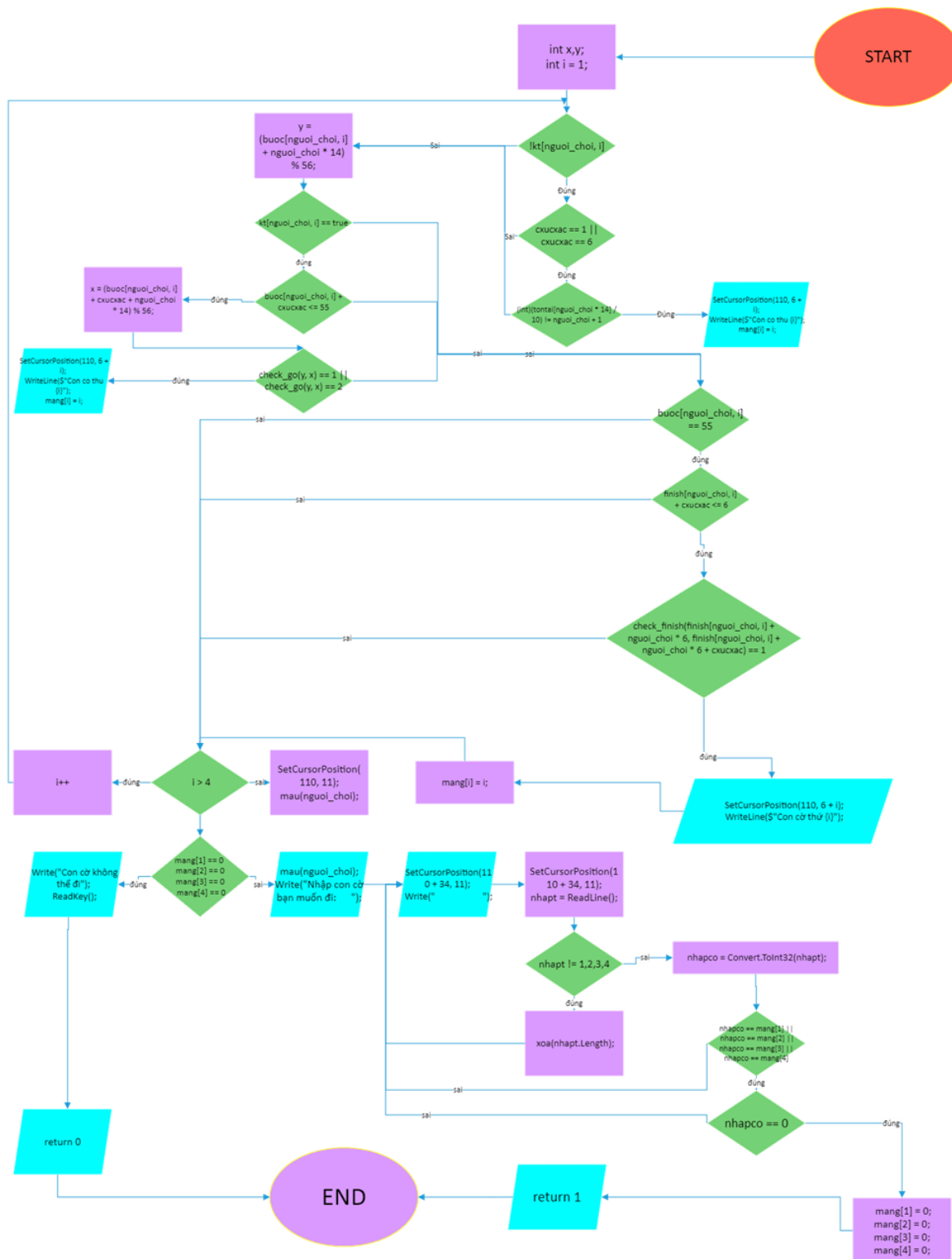
11. Hàm check_finish(y,x)



12. Hàm gogo(x,stt)



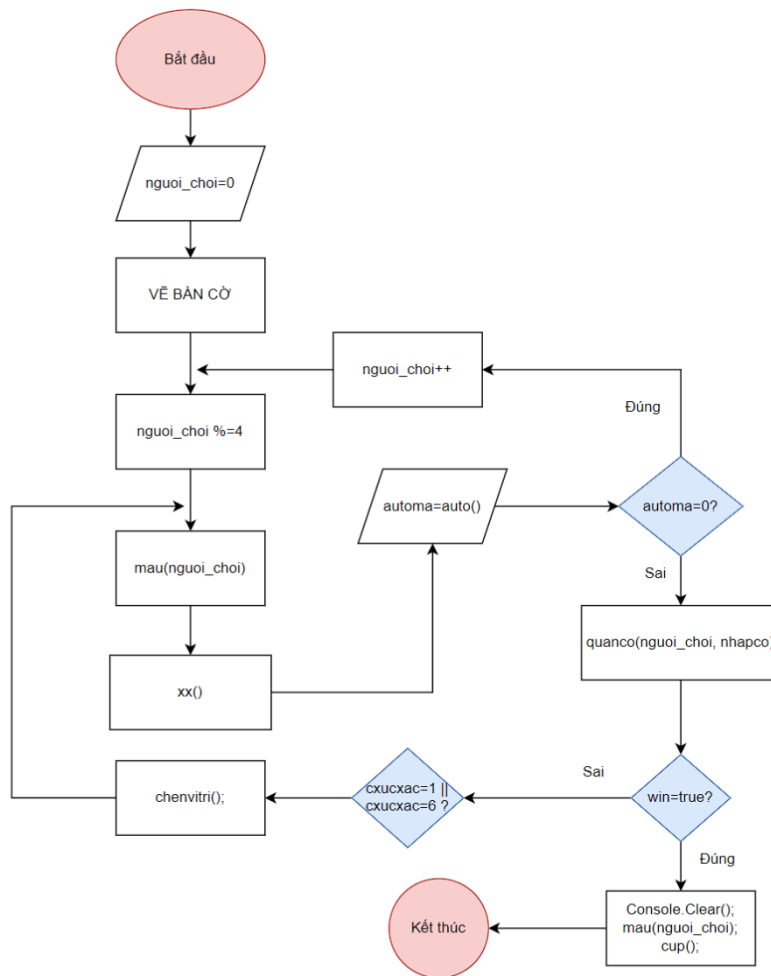
13. Hàm auto()



15

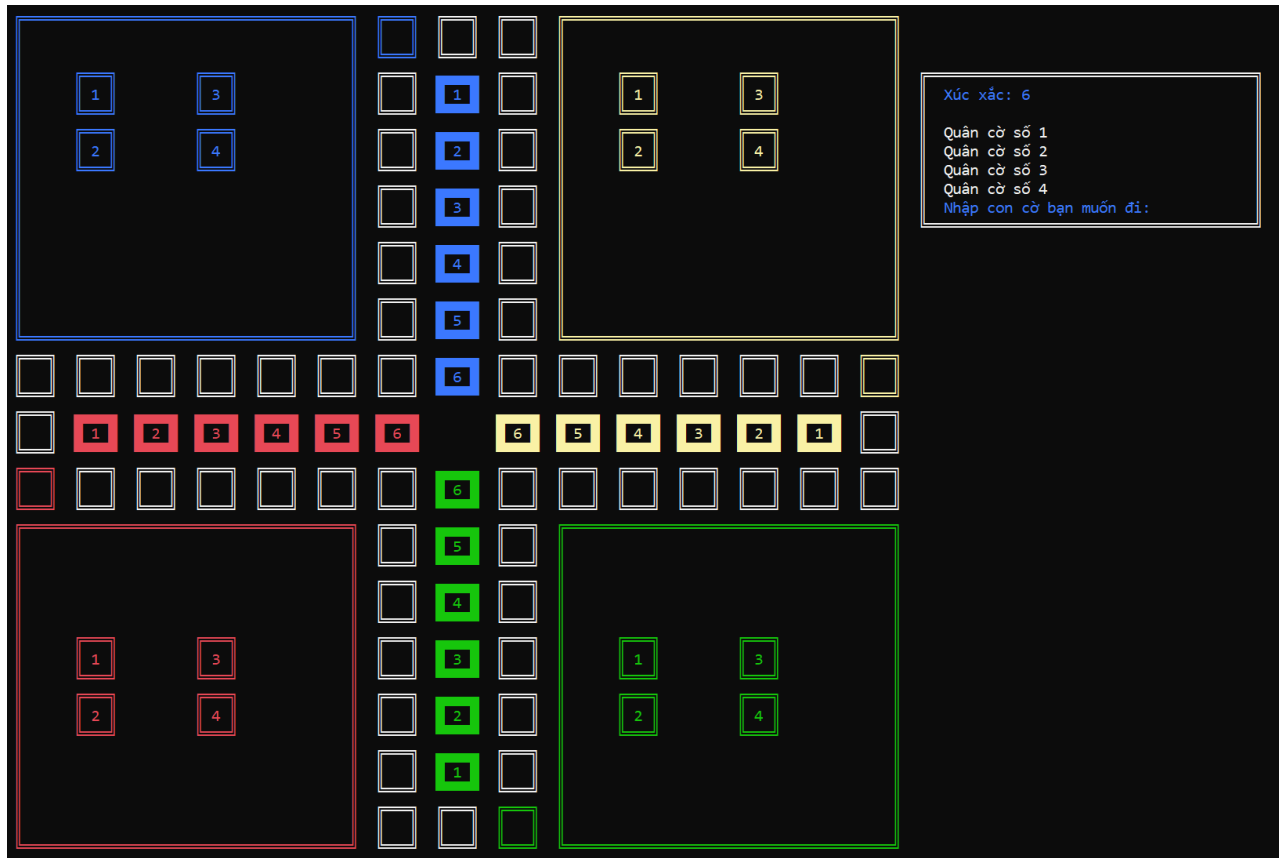


15. Hàm Main

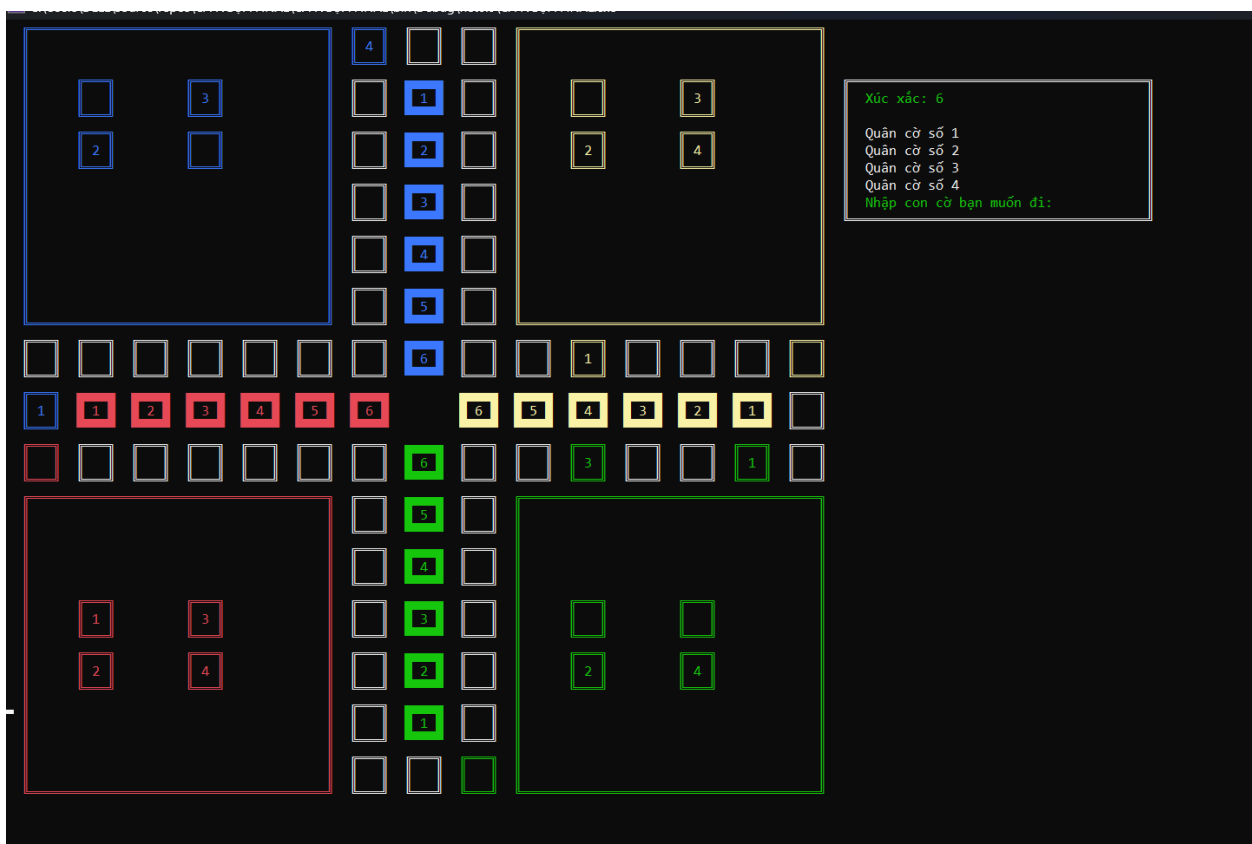
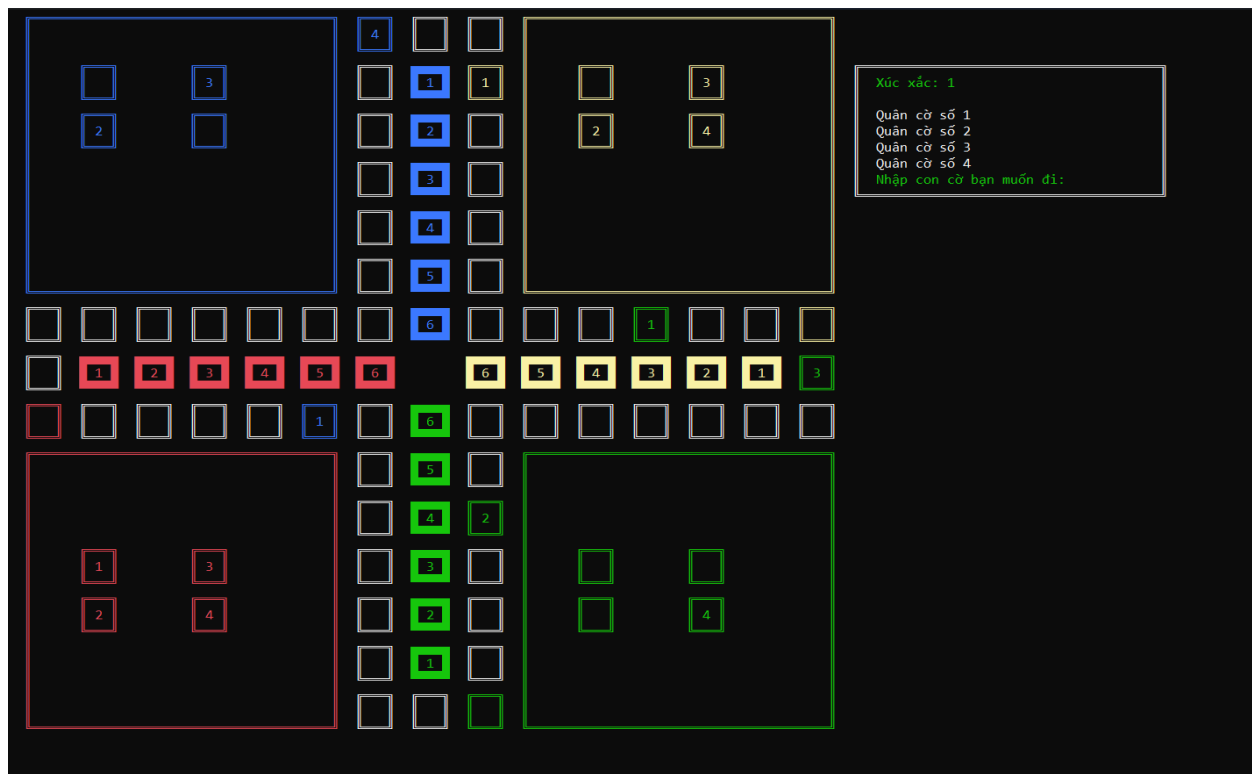


IV. Chạy demo và code

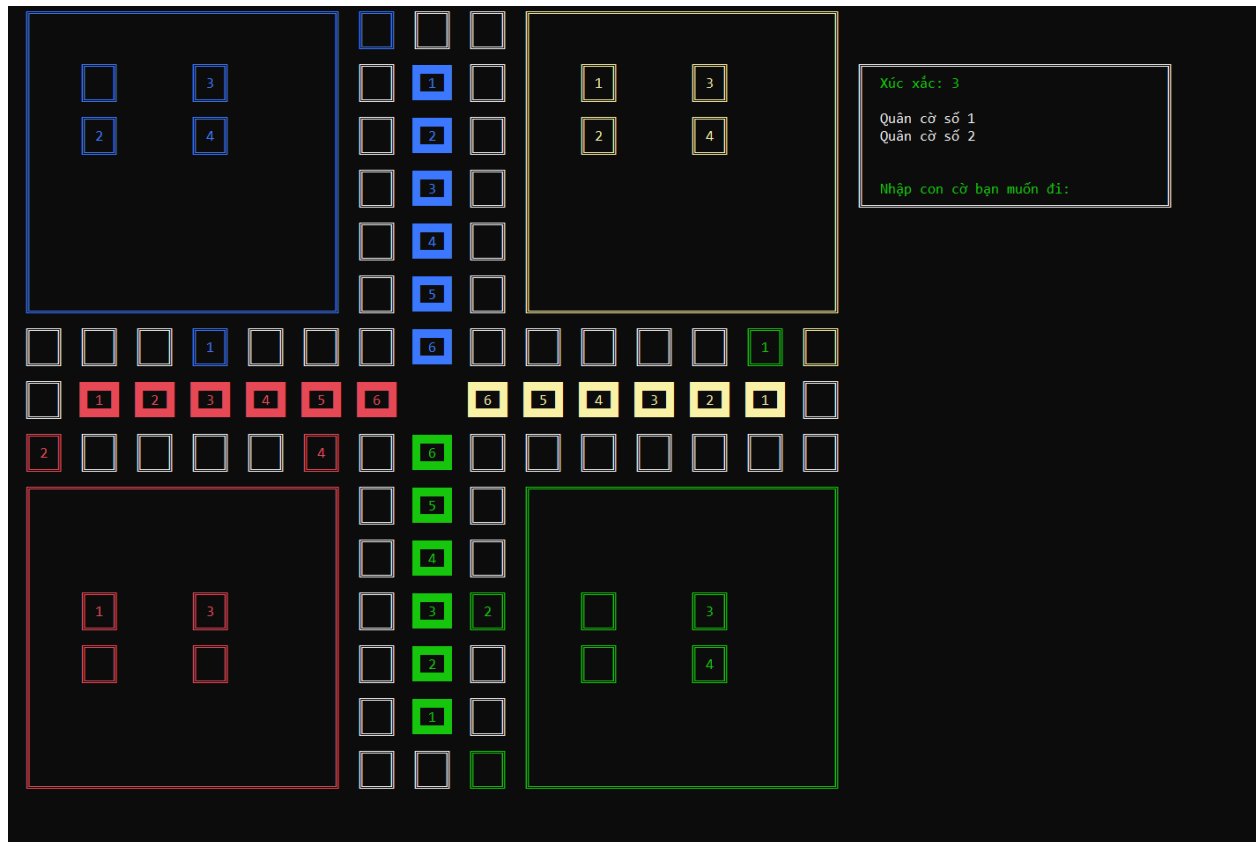
1. Chạy demo
 - Màn hình chính của game cờ cá ngựa.



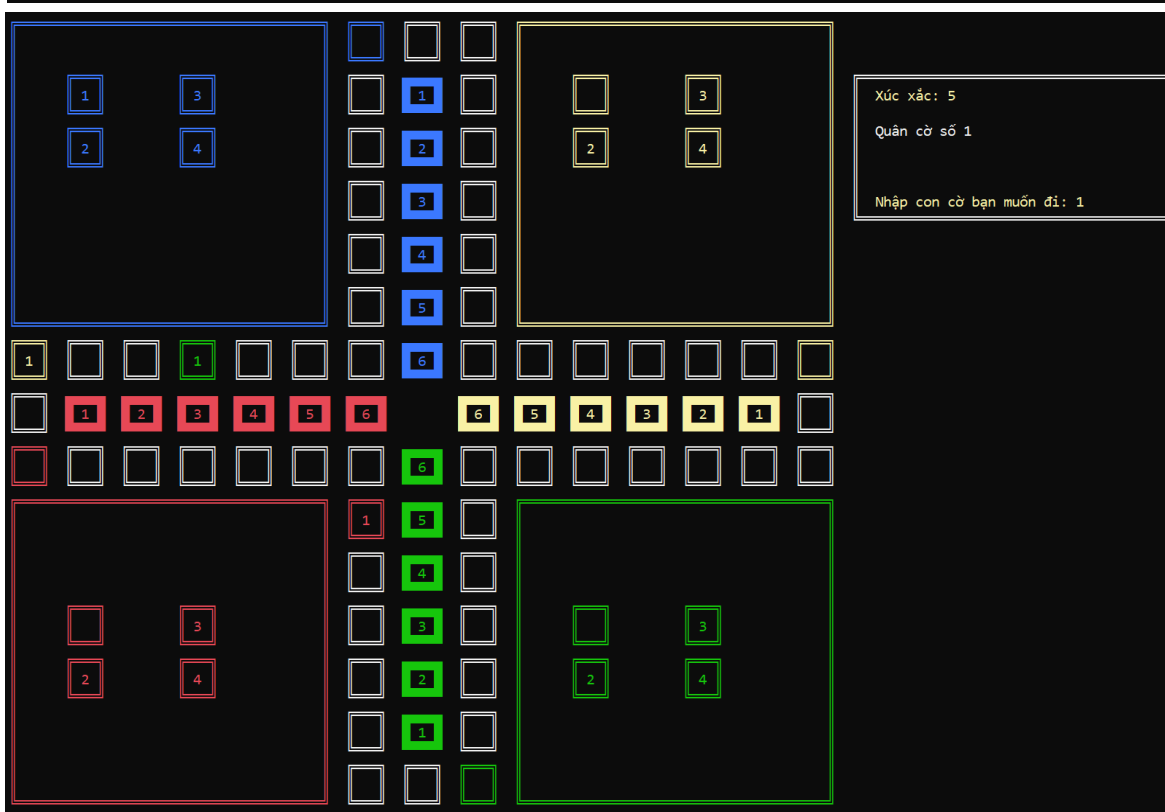
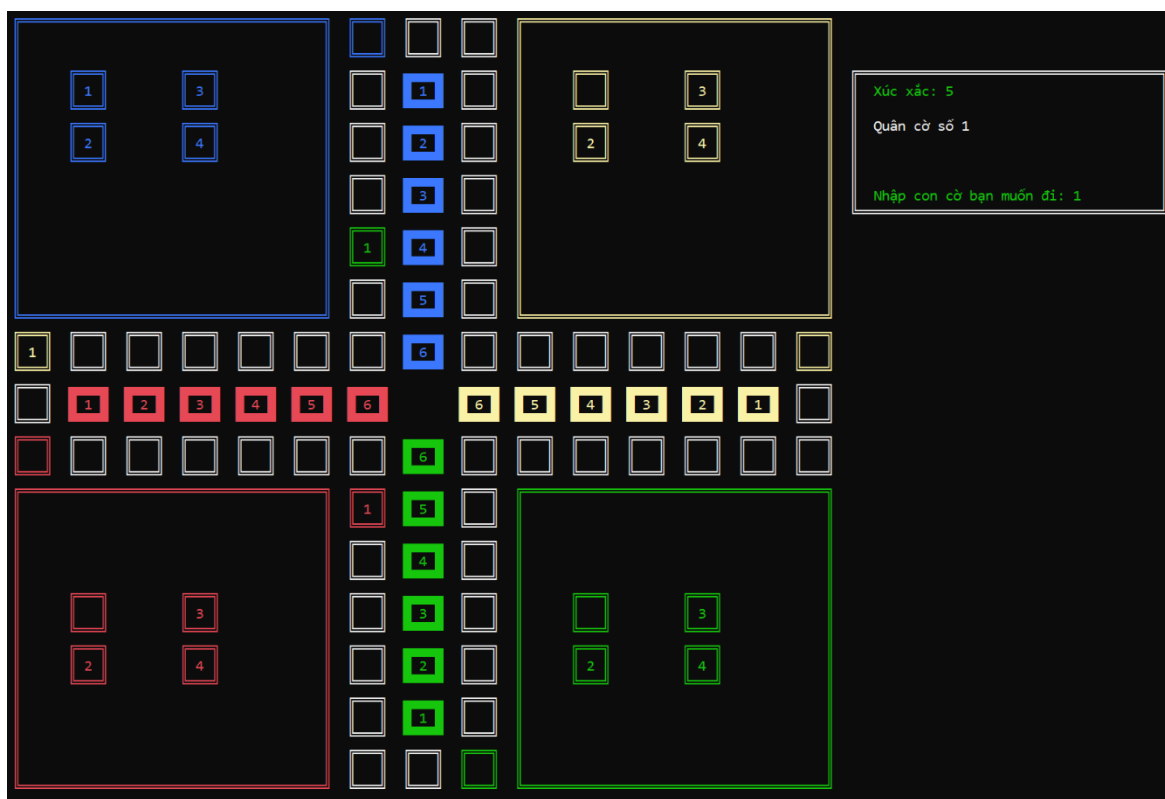
- Khi xúc xúc tung ra bằng 1 hoặc 6 thì con cờ được ra quân và chọn con cờ mà bạn muốn đi.



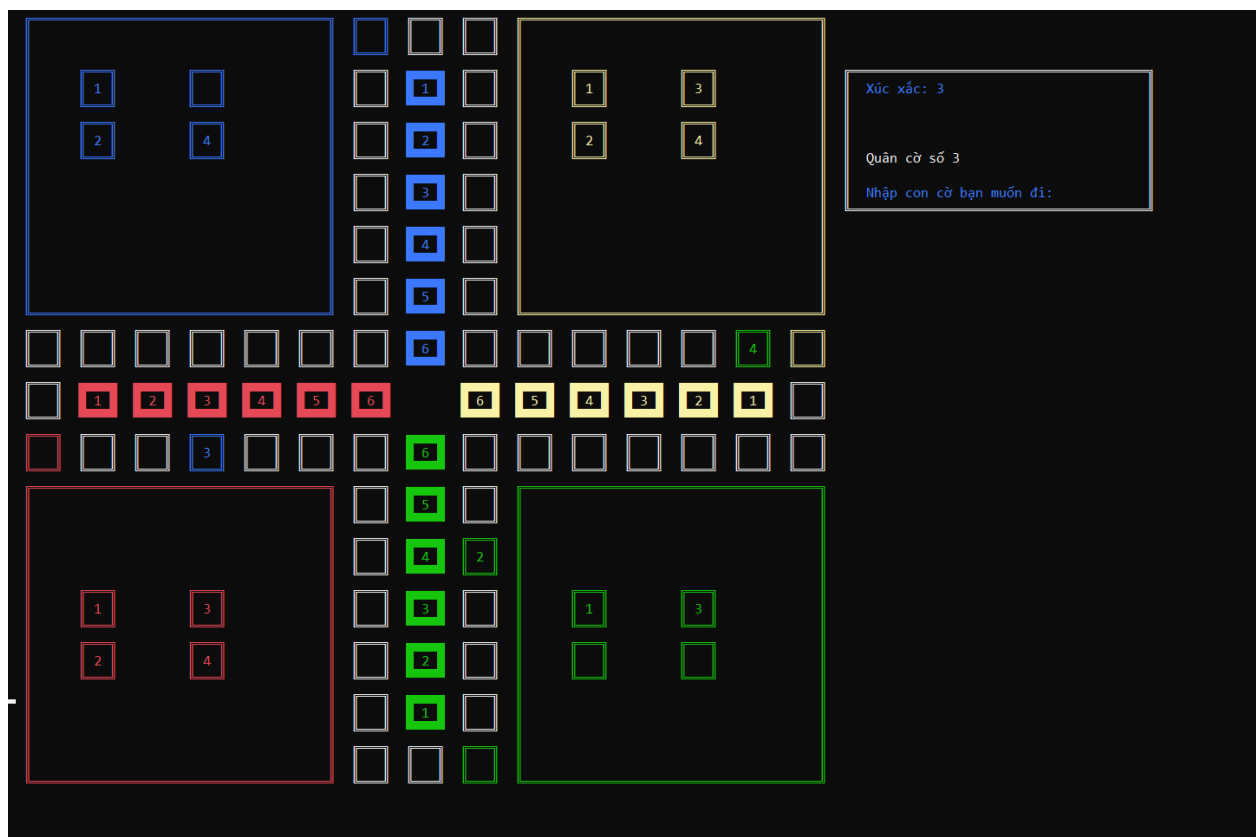
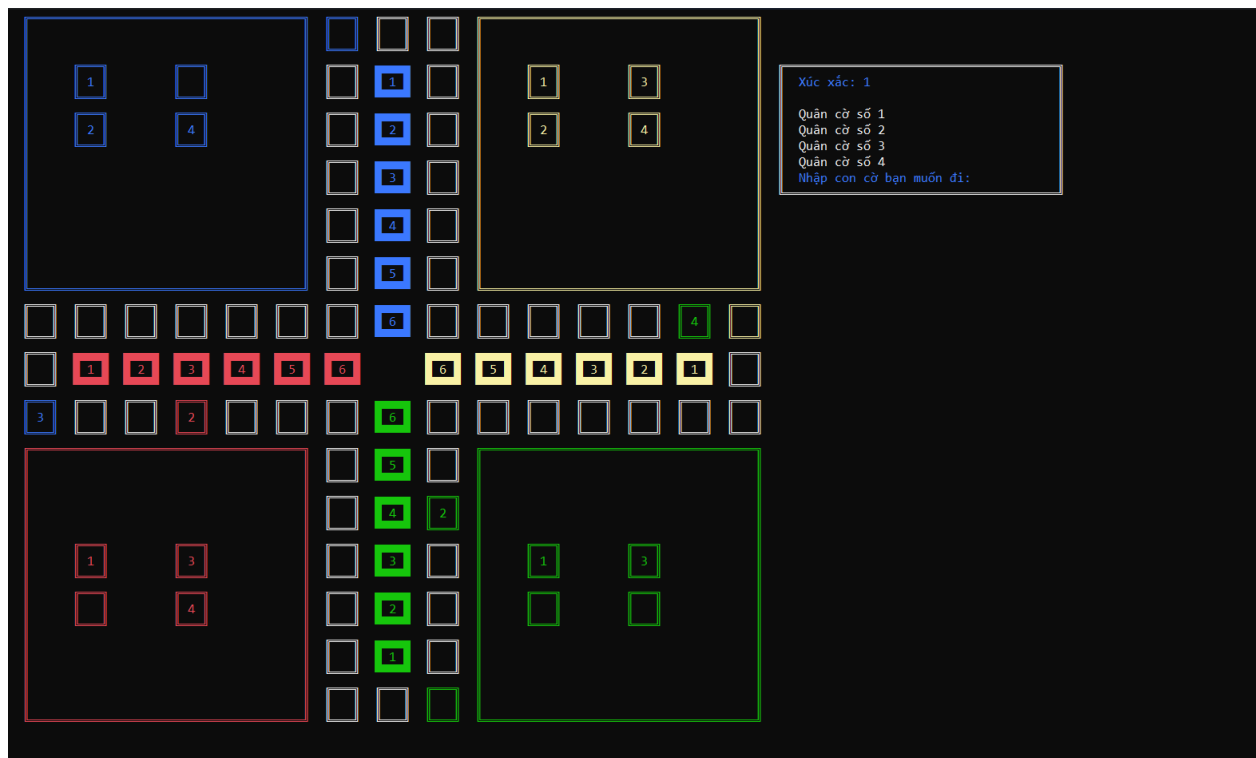
Khi chọn được con cờ bạn muốn đi xúc xúc tiếp tục được tung ra và hệ thống sẽ đưa ra những lựa chọn cho biết có thể chọn con cờ nào để đi tiếp.



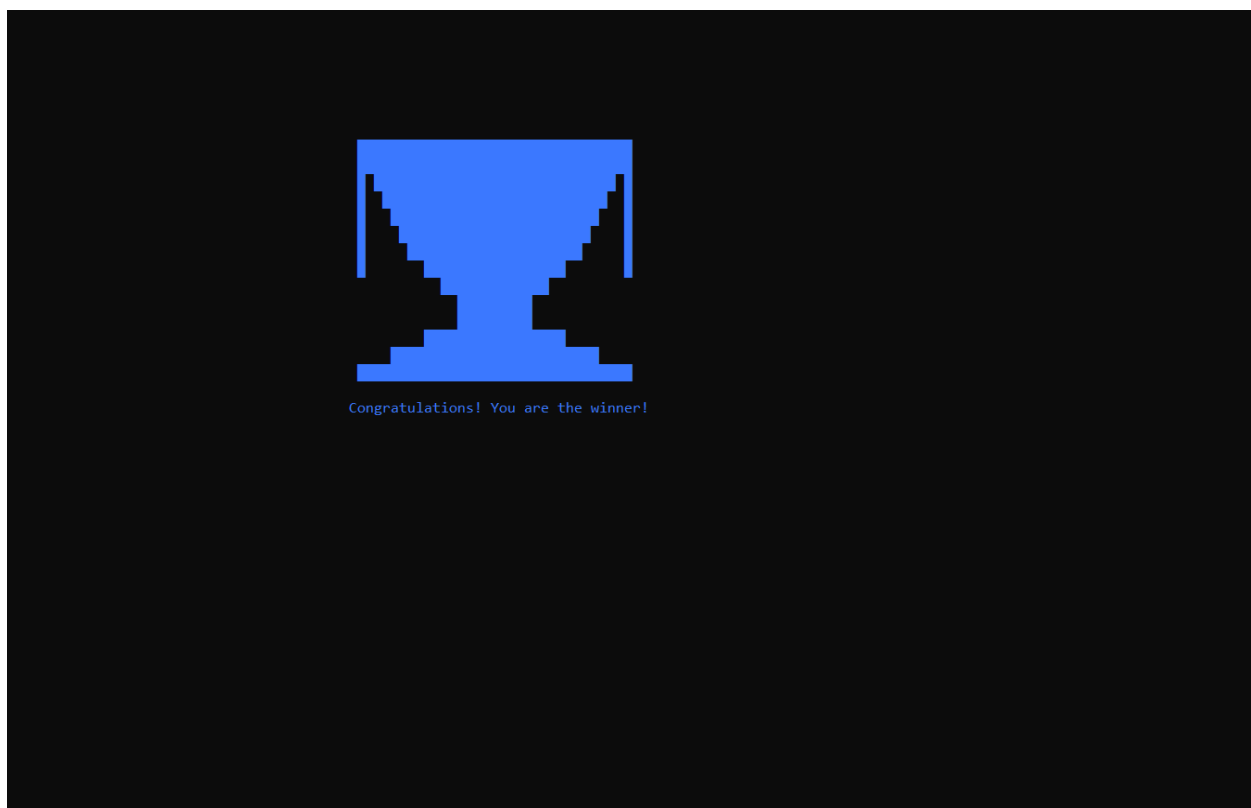
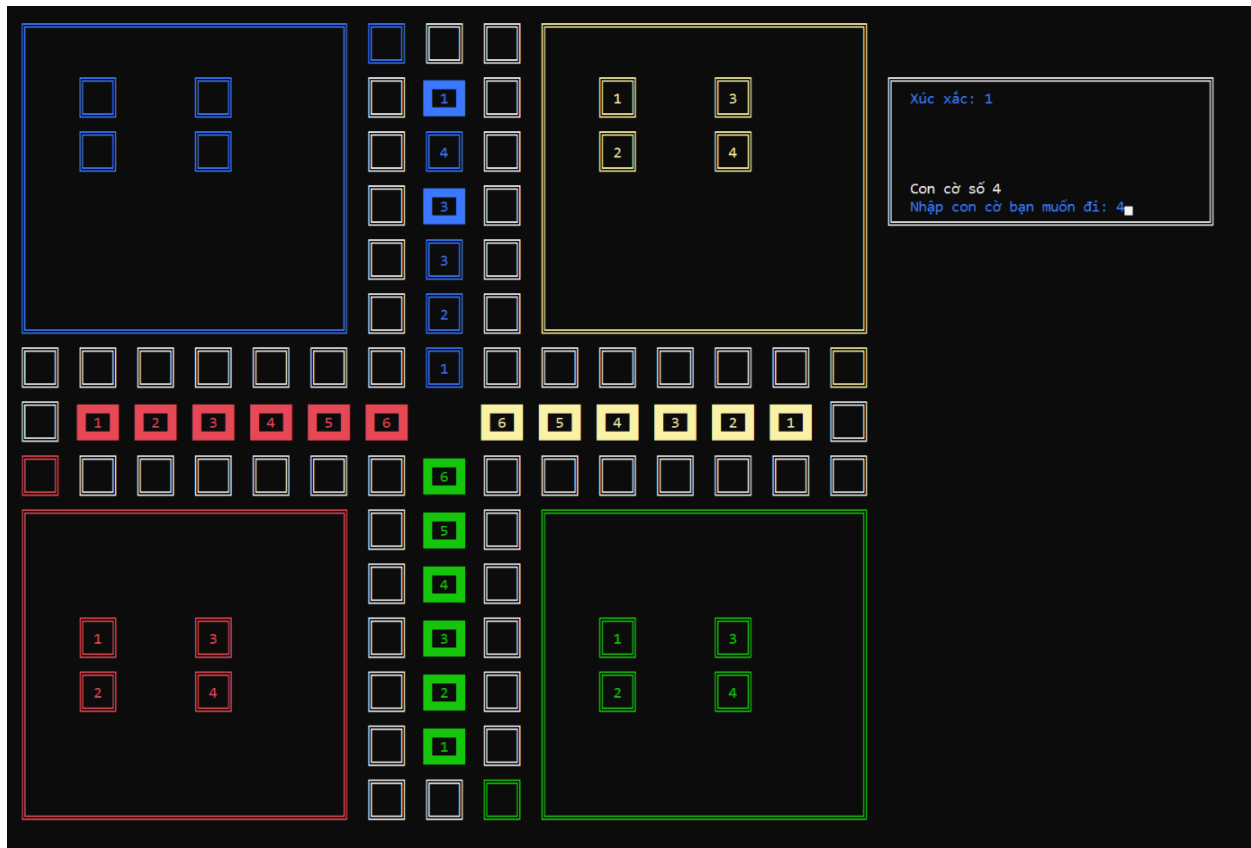
- Khi hết lượt bạn hệ thống sẽ chuyển lượt cho những người chơi khác và tương tự như vậy cho đến khi tìm ra người thắng trò chơi.



- Đá ngựa về chuồng.



- Sau khi các con ngựa đã lên hết đầu tiên thì thắng game này.



2. Code

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Co_ca_ngua
{
    internal class Program
    {
        #region mảng
        //Kiểm tra đã có màu cờ nào win hay chưa
        static public bool win = false;
        static public int nguoi_choi = 0, nhapco;
        static public int cxucxac;

        static public string nhapt;
        static public int[] mang = new int[5];

        static public string[] vitri =
            {
                "6^0", "6^1", "6^2", "6^3", "6^4", "6^5", "6^6", "5^6", "4^6", "3^6", "2^6",
                "1^6", "0^6", "0^7", "0^8", "1^8", "2^8", "3^8", "4^8", "5^8", "6^8", "6^9", "6^10", "6^11", "6^12",
                "6^13", "6^14", "7^14", "8^14", "8^13", "8^12", "8^11", "8^10", "8^9", "8^8", "9^8", "10^8", "11^8",
                "12^8", "13^8", "14^8", "14^7", "14^6", "13^6", "12^6", "11^6", "10^6", "9^6", "8^6", "8^5", "8^4",
                "8^3", "8^2", "8^1", "8^0", "7^0"
            };

        static string[] lentroi =
            { "", "7^1", "7^2", "7^3", "7^4", "7^5", "7^6", "1^7", "2^7", "3^7", "4^7", "5^7", "6^7",
                "7^13", "7^12", "7^11", "7^10", "7^9", "7^8", "13^7", "12^7", "11^7", "10^7", "9^7", "8^7" };
    }
}
```

```
static string[] vtchuong = { "", "1^1", "1^2", "3^1", "3^2", "6^0", "1^11", "1^12", "3^11",
"3^12", "0^8", "10^11", "10^12", "12^11", "12^12", "8^14", "10^1", "10^2", "12^1", "12^2", "14^6"
};
```

```
static public int[] tontai = new int[60]; // 0: không tồn tại cờ, ab: con cờ màu a và stt b tồn
tại ở ô cờ này
```

```
static public bool[,] kt = new bool[4, 5];
static public int[,] buoc = new int[4, 5];
static int[,] finish = new int[5, 7]; // đếm số bước khi về đích
static int[] tontaifi = new int[30];
#endregion
```

```
#region hàm vẽ bàn cờ và xúc xắc
```

```
//Vẽ ô đường đi và quân cờ
```

```
static void oco(string a, int i, int j)
{
    Console.SetCursorPosition(i * 7 + 2, j * 3 + 1);
    Console.WriteLine(" ┌────────┐");
    Console.SetCursorPosition(i * 7 + 2, j * 3 + 2);
    Console.WriteLine($" │ {a} │");
    Console.SetCursorPosition(i * 7 + 2, j * 3 + 3);
    Console.WriteLine(" └────────┘");
}
```

```
// Vẽ ô về đích
```

```
static void cellfinish(string a, int i, int j)
{
    Console.SetCursorPosition(i * 7 + 2, j * 3 + 1);
    Console.WriteLine(" ██████████");
    Console.SetCursorPosition(i * 7 + 2, j * 3 + 2);
    Console.WriteLine($" █ {a} █");
    Console.SetCursorPosition(i * 7 + 2, j * 3 + 3);
    Console.WriteLine(" ██████████");
}
```

[illegible]

```

8);
    Console.SetCursorPosition(Console.WindowWidth / 4, Console.WindowHeight / 5 +
9);
    Console.WriteLine("          ");
    Console.SetCursorPosition(Console.WindowWidth / 4, Console.WindowHeight / 5 +
10);
    Console.WriteLine("          ");
    Console.SetCursorPosition(Console.WindowWidth / 4, Console.WindowHeight / 5 +
11);
    Console.WriteLine("          ");
    Console.SetCursorPosition(Console.WindowWidth / 4, Console.WindowHeight / 5 +
12);
    Console.WriteLine("          ");
    Console.SetCursorPosition(Console.WindowWidth / 4, Console.WindowHeight / 5 +
13);
    Console.WriteLine("          ");
    Console.SetCursorPosition(Console.WindowWidth / 4 - 1, Console.WindowHeight / 5
+ 15);
    Console.WriteLine("Congratulations! You are the winner!");
}

```

#endregion

```

#region hàm kiểm tra quân cờ
static int check_go(int y, int x)
{

    for (int i = 1; i < cxucxac; i++)
    {
        if (tontai[(y + i) % 56] != 0)
        {
            return 0;

```

```

    }
}
int mau_co1 = tontai[y] / 10;
int mau_co2 = tontai[x] / 10;
if (tontai[x] == 0) return 1;
else
{
    if (mau_co1 == mau_co2)
    {
        return 0;
    }
    else if (mau_co1 != mau_co2)
    {

        return 2;
    }
}
return -1;
}
static int check_finish(int y, int x)
{
    for (int i = y + 1; i <= x; i++)
    {
        if (tontai[i] != 0) return 0;
    }
    return 1;
}
static void ve_chuong(int mau_co, int stt)
{
    switch (mau_co)
    {
        case 0: // xanh duong

```

```

        if (stt == 1) oco("1", 1, 1);
        if (stt == 2) oco("2", 1, 2);
        if (stt == 3) oco("3", 3, 1);
        if (stt == 4) oco("4", 3, 2);
        break;
    case 1: // do
        if (stt == 1) oco("1", 1, 11);
        if (stt == 2) oco("2", 1, 12);
        if (stt == 3) oco("3", 3, 11);
        if (stt == 4) oco("4", 3, 12);
        break;
    case 2: // xanh la
        if (stt == 1) oco("1", 10, 11);
        if (stt == 2) oco("2", 10, 12);
        if (stt == 3) oco("3", 12, 11);
        if (stt == 4) oco("4", 12, 12);
        break;
    case 3: // vang
        if (stt == 1) oco("1", 10, 1);
        if (stt == 2) oco("2", 10, 2);
        if (stt == 3) oco("3", 12, 1);
        if (stt == 4) oco("4", 12, 2);
        break;
    }
}

static void back_home(int x)
{
    int mau_co = tontai[x] / 10 - 1;
    int stt = tontai[x] % 10;
    buoc[mau_co, stt] = 0;
    kt[mau_co, stt] = false;
    mau(mau_co);
}

```



```

    ve_chuong(mau_co, stt);

}
static void gogo(int x, int stt)
{
    string[] toadocu = vitri[x].Split('^');
    switch (stt)
    {
        case 1:
            oco("1", Convert.ToInt32(toadocu[0]), Convert.ToInt32(toadocu[1]));
            break;
        case 2:
            oco("2", Convert.ToInt32(toadocu[0]), Convert.ToInt32(toadocu[1]));
            break;
        case 3:
            oco("3", Convert.ToInt32(toadocu[0]), Convert.ToInt32(toadocu[1]));
            break;
        case 4:
            oco("4", Convert.ToInt32(toadocu[0]), Convert.ToInt32(toadocu[1]));
            break;
    }
}
static void gofinish(int x, int stt)
{
    string[] toadocu = lentroi[x].Split('^');
    switch (stt)
    {
        case 1:
            oco("1", Convert.ToInt32(toadocu[0]), Convert.ToInt32(toadocu[1]));
            break;
        case 2:
            oco("2", Convert.ToInt32(toadocu[0]), Convert.ToInt32(toadocu[1]));

```

```

        break;
    case 3:
        oco("3", Convert.ToInt32(toadocu[0]), Convert.ToInt32(toadocu[1]));
        break;
    case 4:
        oco("4", Convert.ToInt32(toadocu[0]), Convert.ToInt32(toadocu[1]));
        break;
    }
}

static void chenvitri()
{
    Console.SetCursorPosition(110, 7);
    Console.WriteLine(" ");
    Console.SetCursorPosition(110, 8);
    Console.WriteLine(" ");
    Console.SetCursorPosition(110, 9);
    Console.WriteLine(" ");
    Console.SetCursorPosition(110, 10);
    Console.WriteLine(" ");
}

static void xoa(int dodai)
{
    for (int i = 1; i <= dodai + 4; i++)
    {
        Console.SetCursorPosition(110 + 25 + i, 11);
        Console.Write(" ");
    }
}

static void mau(int mau)
{
    switch (mau)
    {

```

```

case 0:
    Console.ForegroundColor = ConsoleColor.Blue;
    break;
case 1:
    Console.ForegroundColor = ConsoleColor.Red;
    break;
case 2:
    Console.ForegroundColor = ConsoleColor.Green;
    break;
case 3:
    Console.ForegroundColor = ConsoleColor.Yellow;
    break;
}
}
#endregion

static void quanco(int nguoi_choi, int stt)
{
    int x, y;
    //Ngua ra chuong
    if ((cxucxac == 1 || cxucxac == 6) && !kt[nguoi_choi, stt] &&
(int)(tontai[nguoi_choi*14] / 10) != nguoi_choi + 1)
    {

        kt[nguoi_choi, stt] = true;
        mau(nguoi_choi);
        string[] td = vttchuong[nguoi_choi * 5 + stt].Split('^');
        oco(" ", Convert.ToInt32(td[0]), Convert.ToInt32(td[1]));
        td = vttchuong[(nguoi_choi + 1) * 5].Split('^');
        oco(Convert.ToString(stt), Convert.ToInt32(td[0]), Convert.ToInt32(td[1]));
        if ((int)(tontai[nguoi_choi * 14] / 10) != 0) back_home(nguoi_choi * 14);
        tontai[nguoi_choi * 14] = (nguoi_choi + 1) * 10 + stt;
    }
}

```

```

    }
    else if (!kt[nguoi_choi, stt])
    {
        return;
    }
    else if (kt[nguoi_choi, stt])
    {
        //Ve o co moi

        y = (buoc[nguoi_choi, stt] + nguoi_choi * 14) % 56;
        int tam = buoc[nguoi_choi, stt] + cxucxac;
        if (tam <= 55) buoc[nguoi_choi, stt] += cxucxac;
        if (tam > 55 && buoc[nguoi_choi, stt] == 55 && (finish[nguoi_choi, stt] + cxucxac
<= 6))
        {
            y = (finish[nguoi_choi, stt] + nguoi_choi * 6);
            x = y + cxucxac;
            int ckgoal = check_finish(y, x);
            if (ckgoal == 1)
            {
                if (y == nguoi_choi * 6)
                {
                    string[] td = vitri[(nguoi_choi * 14 - 1 + 56) % 56].Split('^');
                    Console.ForegroundColor = ConsoleColor.White;
                    oco(" ", Convert.ToInt32(td[0]), Convert.ToInt32(td[1]));
                    tontai[(nguoi_choi * 14 - 1 + 56) % 56] = 0;
                }
            }
            else
            {
                string xoafi = (y - 6 * nguoi_choi).ToString();
                string[] toadocu = lentroi[y].Split('^');

```

```

        tontaifi[y] = 0;
        mau(nguoi_choi);
        cellfinish(xoafi, Convert.ToInt32(toadocu[0]), Convert.ToInt32(toadocu[1]));
    }
    mau(nguoi_choi);
    gofinish(x, stt);
    tontaifi[x] = (nguoi_choi + 1) * 10 + stt;
    finish[nguoi_choi, stt] += cxucxac;
    if (tontaifi[nguoi_choi * 6 + 6] != 0 && tontaifi[nguoi_choi * 6 + 5] != 0 &&
tontaifi[nguoi_choi * 6 + 4] != 0 && tontaifi[nguoi_choi * 6 + 3] != 0)
    {
        win = true;
    }
}
return;
}

// Trường hợp con cờ ở ô 54 và xx > 1 thì không vào được if trên và cũng không return
=> lỗi. Do đó cần if này
if (tam > 55)
{
    return;
}
x = (buoc[nguoi_choi, stt] + nguoi_choi * 14) % 56;
int ckg = check_go(y, x);
if (ckg == 0)
{
    buoc[nguoi_choi, stt] -= cxucxac;
    return;
}
if (ckg == 2)
{
    mau(nguoi_choi);

```

```

        gogo(x, stt);
        string vtcu = vitri[y];
        string[] toadocu = vtcu.Split('^');
        if (Convert.ToInt32(toadocu[0]) == 6 && Convert.ToInt32(toadocu[1]) == 0)
Console.ForegroundColor = ConsoleColor.Blue;
        else if (Convert.ToInt32(toadocu[0]) == 0 && Convert.ToInt32(toadocu[1]) == 8)
Console.ForegroundColor = ConsoleColor.Red;
        else if (Convert.ToInt32(toadocu[0]) == 8 && Convert.ToInt32(toadocu[1]) == 14)
Console.ForegroundColor = ConsoleColor.Green;
        else if (Convert.ToInt32(toadocu[0]) == 14 && Convert.ToInt32(toadocu[1]) ==
6) Console.ForegroundColor = ConsoleColor.Yellow;
        else Console.ForegroundColor = ConsoleColor.White;
        oco(" ", Convert.ToInt32(toadocu[0]), Convert.ToInt32(toadocu[1]));
        back_home(x);
        tontai[x] = (nguoi_choi + 1) * 10 + stt;
        tontai[y] = 0;
    }
    if (ckg == 1)
    {
        mau(nguoi_choi);
        gogo(x, stt);
        string vtcu = vitri[y];
        string[] toadocu = vtcu.Split('^');
        if (Convert.ToInt32(toadocu[0]) == 6 && Convert.ToInt32(toadocu[1]) == 0)
Console.ForegroundColor = ConsoleColor.Blue;
        else if (Convert.ToInt32(toadocu[0]) == 0 && Convert.ToInt32(toadocu[1]) == 8)
Console.ForegroundColor = ConsoleColor.Red;
        else if (Convert.ToInt32(toadocu[0]) == 8 && Convert.ToInt32(toadocu[1]) == 14)
Console.ForegroundColor = ConsoleColor.Green;
        else if (Convert.ToInt32(toadocu[0]) == 14 && Convert.ToInt32(toadocu[1]) ==
6) Console.ForegroundColor = ConsoleColor.Yellow;
        else Console.ForegroundColor = ConsoleColor.White;
        oco(" ", Convert.ToInt32(toadocu[0]), Convert.ToInt32(toadocu[1]));
        tontai[x] = (nguoi_choi + 1) * 10 + stt;
    }
}

```

```

        tontai[y] = 0;
    }
}
}
static int auto()
{
    chenvitri();
    int y, x;
    Console.ForegroundColor = ConsoleColor.White;
    for (int i = 1; i <= 4; i++)
    {
        if ((cxucxac == 1 || cxucxac == 6) && !kt[nguai_choi, i] && (int)(tontai[nguai_choi
* 14] / 10) != nguoi_choi + 1)
        {
            Console.SetCursorPosition(110, 6 + i);
            Console.WriteLine($"Quân cờ số {i}");
            mang[i] = i;
        }
        else
        {
            y = (buoc[nguai_choi, i] + nguoi_choi * 14) % 56;
            if (buoc[nguai_choi, i] + cxucxac <= 55 && kt[nguai_choi, i] == true)
            {
                x = (buoc[nguai_choi, i] + cxucxac + nguoi_choi * 14) % 56;
                if (check_go(y, x) == 1 || check_go(y, x) == 2)
                {
                    Console.SetCursorPosition(110, 6 + i);
                    Console.WriteLine($"Quân cờ số {i}");
                    mang[i] = i;
                }
            }
            else if (buoc[nguai_choi, i] == 55 && (finish[nguai_choi, i] + cxucxac <= 6))

```

```

        {
            if (check_finish(finish[nguai_choi, i] + nguoi_choi * 6, finish[nguai_choi, i] +
nguai_choi * 6 + cxucxac) == 1)
            {
                Console.SetCursorPosition(110, 6 + i);
                Console.WriteLine($"Con cờ số {i}");
                mang[i] = i;
            }
        }
    }
}
Console.SetCursorPosition(110, 11);
mau(nguai_choi);
if (mang[1] == 0 && mang[2] == 0 && mang[3] == 0 && mang[4] == 0)
{
    Console.Write("Quân cờ không thể đi      ");
    Console.ReadKey();
    return 0;
}
mau(nguai_choi);
Console.Write("Nhập con cờ bạn muốn đi: ");
nhaplaixd:
    Console.SetCursorPosition(110 + 25, 11);
    Console.Write("      ");
    Console.SetCursorPosition(110 + 25, 11);
    nhapt = Console.ReadLine();
    //Random rand = new Random();
    //nhapt = rand.Next(1, 5).ToString();
    if (nhapt != "4" && nhapt != "1" && nhapt != "2" && nhapt != "3")
    {
        xoa(nhapt.Length);
    }
}

```



```

        goto nhaplaixd;
    }
    nhapco = Convert.ToInt32(nhapt);
    if (nhapco == mang[1] || nhapco == mang[2] || nhapco == mang[3] || nhapco == mang[4])
    {
        if (nhapco == 0)
        {
            goto nhaplaixd;
        }
        mang[1] = 0; mang[2] = 0; mang[3] = 0; mang[4] = 0;
    }
    else
    {
        goto nhaplaixd;
    }
    return 1;
}

static void Main(string[] args)
{
    Console.OutputEncoding = Encoding.Unicode;
    #region Banco
    int a = 0; string b;
    //--VE O CO XANH DUONG
    for (int i = 6; i <= 8; i++)
        for (int j = 0; j <= 6; j++)
        {
            if (i == 7 && j > 0)
            {
                Console.ForegroundColor = ConsoleColor.Blue;
                a++;
                b = Convert.ToString(a);
                cellfinish(b, i, j);
            }
        }
    }
}

```

```

    }
    else
    {
        Console.ForegroundColor = ConsoleColor.White;
        b = " ";
        oco(b, i, j);
    }
    if (i == 6 && j == 0)
    {
        Console.ForegroundColor = ConsoleColor.Blue;
        b = " ";
        oco(b, i, j);
    }

}

//Ve chuong ngua
for (int j = 0 * 3 + 1; j <= 6 * 3; j++)
{
    for (int i = 9 * 7 + 2; i <= (14 + 1) * 7 - 1; i++)
    {
        Console.SetCursorPosition(i, j);
        Console.ForegroundColor = ConsoleColor.Yellow;
        if (j == 0 * 3 + 1 || j == 6 * 3) Console.Write("=");
        else if (i == 9 * 7 + 2 || i == (14 + 1) * 7 - 1) Console.Write("||");
        else Console.Write(" ");
    }
}

//Ve 4 goc
Console.SetCursorPosition(9 * 7 + 2, 0 * 3 + 1); Console.Write("┌");
Console.SetCursorPosition(9 * 7 + 2, 6 * 3); Console.Write("└");
Console.SetCursorPosition((14 + 1) * 7 - 1, 0 * 3 + 1); Console.Write("┐");
Console.SetCursorPosition((14 + 1) * 7 - 1, 6 * 3); Console.Write("┘");

```

```

//--VE O CO DO
a = 0;
for (int i = 6; i <= 9; i++)
    for (int j = 0; j <= 6; j++)
    {

        if (i == 7 && j > 0)
        {
            Console.ForegroundColor = ConsoleColor.Red;
            a++;
            b = Convert.ToString(a);
            cellfinish(b, j, i);
        }
        else
        {
            Console.ForegroundColor = ConsoleColor.White;
            b = " ";
            oco(b, j, i);
        }
        if (i == 8 && j == 0)
        {
            Console.ForegroundColor = ConsoleColor.Red;
            b = " ";
            oco(b, j, i);
        }

    }

//Ve chuong ngua
for (int j = 0 * 3 + 1; j <= 6 * 3; j++)
{
    for (int i = 2; i <= 41; i++)

```

```

{
    Console.SetCursorPosition(i, j);
    Console.ForegroundColor = ConsoleColor.Blue;
    if (j == 1 || j == 18) Console.Write("=");
    else if (i == 2 || i == 41) Console.Write("||");
    else Console.Write(" ");
}
}
//ve 4 goc
Console.SetCursorPosition(2, 1); Console.Write("┌");
Console.SetCursorPosition(2, 18); Console.Write("└");
Console.SetCursorPosition(41, 1); Console.Write("┐");
Console.SetCursorPosition(41, 18); Console.Write("┘");
//--VE O CO XANH LA
a = 7;
for (int i = 8; i <= 14; i++)
    for (int j = 6; j <= 8; j++)
    {

        if (j == 7 && i > 7 && i < 14)
        {
            Console.ForegroundColor = ConsoleColor.Green;
            a--;
            b = Convert.ToString(a);
            cellfinish(b, j, i);
        }
        else
        {
            Console.ForegroundColor = ConsoleColor.White;
            b = " ";
            oco(b, j, i);
        }
    }
}

```

```

        if (i == 14 && j == 8)
        {
            Console.ForegroundColor = ConsoleColor.Green;
            b = " ";
            oco(b, j, i);
        }

    }

//Ve chuong ngua
for (int j = 9 * 3 + 1; j <= 15 * 3; j++)
{
    for (int i = 2; i <= 41; i++)
    {
        Console.SetCursorPosition(i, j);
        Console.ForegroundColor = ConsoleColor.Red;
        if (j == 9 * 3 + 1 || j == 15 * 3) Console.Write("=");
        else if (i == 2 || i == 41) Console.Write("||");
        else Console.Write(" ");
    }
}

//ve 4 goc
Console.SetCursorPosition(2, 9 * 3 + 1); Console.Write("┌");
Console.SetCursorPosition(2, 15 * 3); Console.Write("└");
Console.SetCursorPosition(41, 9 * 3 + 1); Console.Write("┐");
Console.SetCursorPosition(41, 15 * 3); Console.Write("┘");
//--VE O CO VANG
a = 7;
for (int i = 8; i <= 14; i++)
    for (int j = 6; j <= 8; j++)
    {

        if (j == 7 && i > 7 && i < 14)

```

```

    {
        Console.ForegroundColor = ConsoleColor.Yellow;
        a--;
        b = Convert.ToString(a);
        cellfinish(b, i, j);
    }
else
{
    Console.ForegroundColor = ConsoleColor.White;
    b = " ";
    oco(b, i, j);
}
if (i == 14 && j == 6)
{
    Console.ForegroundColor = ConsoleColor.Yellow;
    b = " ";
    oco(b, i, j);
}

}

//Ve chuong ngua
for (int j = 9 * 3 + 1; j <= 15 * 3; j++)
{
    for (int i = 65; i <= 104; i++)
    {
        Console.SetCursorPosition(i, j);
        Console.ForegroundColor = ConsoleColor.Green;
        if (j == 9 * 3 + 1 || j == 15 * 3) Console.Write("=");
        else if (i == 65 || i == 104) Console.Write("||");
        else Console.Write(" ");
    }
}
}

```

```

//ve 4 goc
Console.SetCursorPosition(65, 9 * 3 + 1); Console.Write("𐀀");
Console.SetCursorPosition(65, 15 * 3); Console.Write("𐀁");
Console.SetCursorPosition(104, 9 * 3 + 1); Console.Write("𐀂");
Console.SetCursorPosition(104, 15 * 3); Console.Write("𐀃");
//ve ngua trong chuong
Console.ForegroundColor = ConsoleColor.Blue;
oco("1", 1, 1);
oco("2", 1, 2);
oco("3", 3, 1);
oco("4", 3, 2);
Console.ForegroundColor = ConsoleColor.Yellow;
oco("1", 10, 1);
oco("2", 10, 2);
oco("3", 12, 1);
oco("4", 12, 2);
Console.ForegroundColor = ConsoleColor.Red;
oco("1", 1, 11);
oco("2", 1, 12);
oco("3", 3, 11);
oco("4", 3, 12);
Console.ForegroundColor = ConsoleColor.Green;
oco("1", 10, 11);
oco("2", 10, 12);
oco("3", 12, 11);
oco("4", 12, 12);

for (int j = 1 * 3 + 1; j <= 4 * 3; j++)
{
    for (int i = 15 * 7 + 2; i <= (20 + 1) * 7 - 1; i++)
    {
        Console.SetCursorPosition(i, j);
    }
}

```

```

        Console.ForegroundColor = ConsoleColor.White;
        if (j == 1 * 3 + 1 || j == 4 * 3) Console.Write("=");
        else if (i == 15 * 7 + 2 || i == (20 + 1) * 7 - 1) Console.Write("||");
        else Console.Write(" ");
    }
}
Console.SetCursorPosition(15 * 7 + 2, 1 * 3 + 1); Console.Write("┐");
Console.SetCursorPosition(15 * 7 + 2, 4 * 3); Console.Write("└");
Console.SetCursorPosition((20 + 1) * 7 - 1, 1 * 3 + 1); Console.Write("┐");
Console.SetCursorPosition((20 + 1) * 7 - 1, 4 * 3); Console.Write("└");
#endregion

while (true)
{
    nguoi_choi %= 4;
ditiep:
    mau(nguoi_choi);
    xx();
    Console.SetCursorPosition(110, 6);
    int automa = auto();
    if (automa == 0) goto thoat;
    quanco(nguoi_choi, nhapco);
    if (win == true) goto end;
thoat:
    if (cxucxac == 1 || cxucxac == 6)
    {
        chenvitri();
        goto ditiep;
    }
    nguoi_choi++;
}
end:

```



```
    Console.Clear();  
    mau(nguoi_choi);  
    cup();  
    Console.ReadLine();  
}  
}  
}
```