

TRƯỜNG ĐẠI HỌC CẦN THƠ

**TRƯỜNG BÁCH KHOA**

**KHOA TỰ ĐỘNG HÓA**

**ĐỒ ÁN MÔN HỌC**

**LẬP TRÌNH CĂN BẢN ĐIỆN TỬ**

**THIẾT KẾ GAME DÒ MÌN ĐƠN GIẢN**

**Nhóm 05 MSSV Giảng viên hướng dẫn:**

Lâm Anh Duy B2204152 Phạm Duy Nghiệp

Nguyễn Quốc Huy B2204168

Nguyễn Văn Hoàng B2204167

Trần Hữu Nghĩa B2204188

Nguyễn Thị Hữu Duyên B2204156

Nguyễn Văn Khang B2204173

Nguyễn Quốc Nguyên B2204190

Cần Thơ, tháng 3 năm 2024

**MỤC LỤC**

[Chương 1. Tổng quan 1](#_Toc162958314)

[1.1. Mục tiêu, yêu cầu (chi tiết) 1](#_Toc162958315)

[1.1.1. Mục tiêu 1](#_Toc162958316)

[1.1.2. Yêu cầu 1](#_Toc162958317)

[1.2. Giới hạn 1](#_Toc162958318)

[1.3. Hướng giải quyết 1](#_Toc162958319)

[Chương 2. Nội dung 2](#_Toc162958320)

[2.1. Giải thuật (lưu đồ) 2](#_Toc162958321)

[2.2. Chương trình chi tiết 17](#_Toc162958322)

[Chương 3. Kết luận 32](#_Toc162958323)

[3.1. Kết quả và hướng dẫn sử dụng 32](#_Toc162958324)

[3.1.1. Kết quả 32](#_Toc162958325)

[3.1.2. Hướng dẫn sử dụng trò chơi 32](#_Toc162958326)

[3.2. Nhận xét (ưu, khuyết điểm, hướng phát triển...) 32](#_Toc162958327)

[Chương 4. Tài liệu tham khảo 34](#_Toc162958328)

# Tổng quan

## Mục tiêu, yêu cầu (chi tiết)

### Mục tiêu

* Phát triển trò chơi Dò mìn với giao diện dễ sử dụng và có các chức năng cơ bản như: chọn độ khó trò chơi, chọn ô, kiểm tra ô có mìn hay không, hiển thị số mìn xung quanh ô, cắm cờ, hiển thị số ô chưa mở, kết luận thắng hay thua, chơi lại và chọn lại độ khó trò chơi, đếm số lần đã chơi.

### Yêu cầu

* Ít nhất phải có 8 công việc liên quan với nhau.
* Ít nhất 8 lệnh điều kiện.
* Ít nhất 8 lệnh lặp.
* In ra màn hình số lần thực hiện chương trình.

## Giới hạn

* Chương trình chỉ tập trung vào việc phát triển trò chơi dò mìn cơ bản mà không xử lý các yếu tố phức tạp như tính toán thời gian chơi, xếp hạng, phát triển đồ họa cao, phát triển trên các nền tảng khác,...

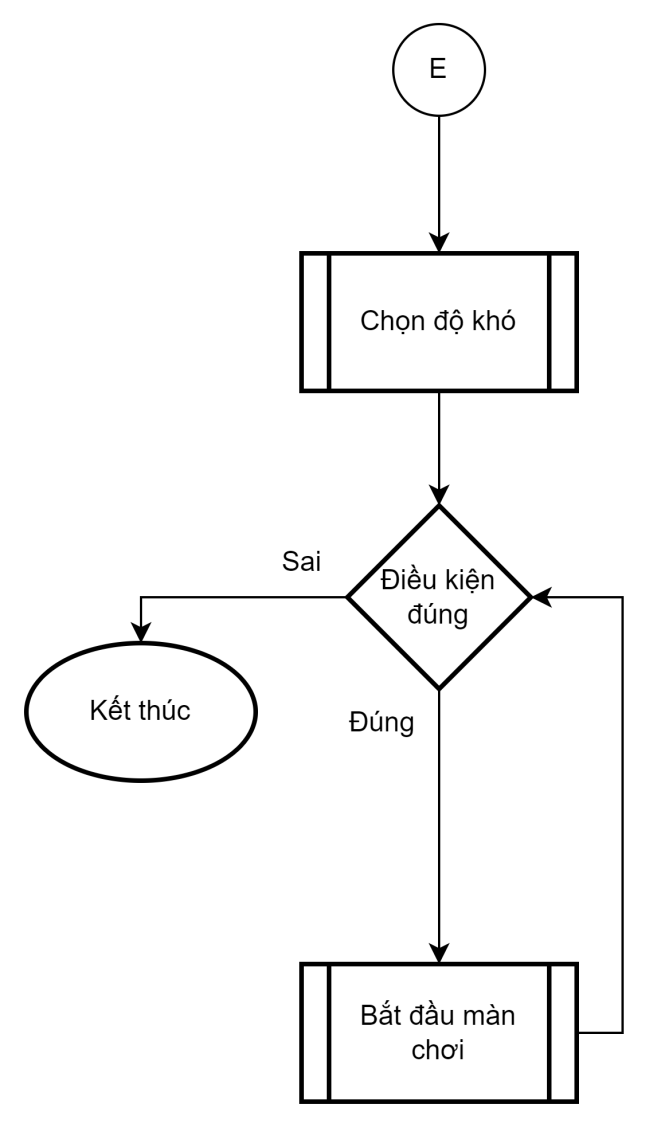
## Hướng giải quyết

* Sử dụng cấu trúc phù hợp để lưu trữ trạng thái của bảng mìn.
* Các hàm để thực hiện các chức năng của trò chơi như:
* ‘demsolan’: Đếm số lần chơi đã được lưu trong tệp và in ra màn hình.
* ‘Tao\_min’: Tạo các ô có mìn trên bản đồ ngẫu nhiên.
* ‘Dem\_min’: Đếm số mìn xung quanh từng ô và gán giá trị đó vào ô trống.
* ‘Mo\_o’: Mở ô khi người chơi chọn một ô.
* ‘printMap’: In ra bản đồ hiện tại của trò chơi.
* ‘So\_o\_con\_lai’: Đếm số ô chưa được mở.
* ‘Kiem\_tra’: Kiểm tra xem người chơi đã thắng, thua, hoặc trò chơi chưa kết thúc.
* ‘Cam\_co’: Đặt hoặc gỡ cờ hiệu trên một ô.
* ‘Chon\_level’: Cho phép người chơi chọn độ khó của trò chơi.
* ‘Choi\_lai’: Cho phép người chơi chọn chơi lại, chọn mức khó mới hoặc thoát khỏi trò chơi.
* ‘Newgame’: Bắt đầu một trò chơi mới với mức độ khó được chọn.

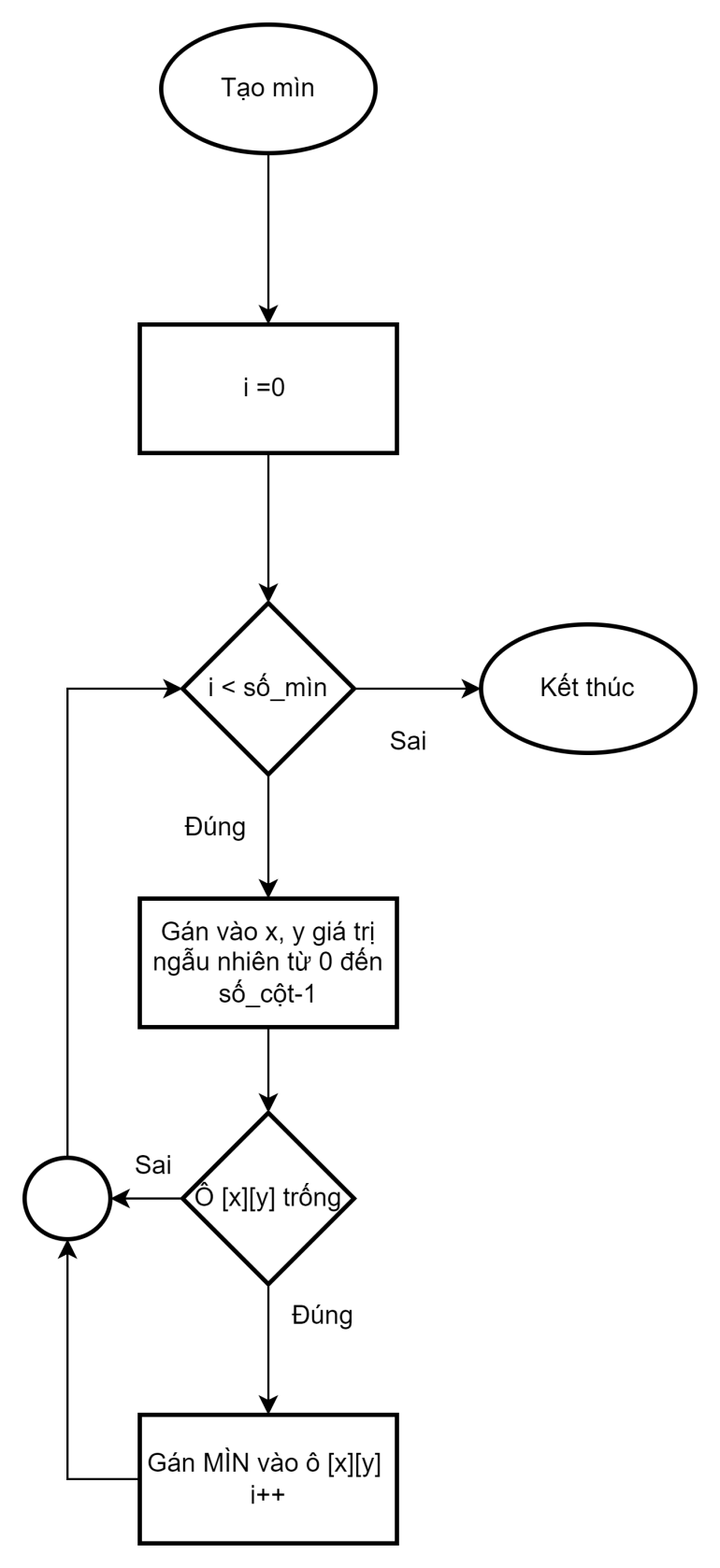
# Nội dung

## Giải thuật (lưu đồ)

* Hàm main



* Hàm tạo mìn

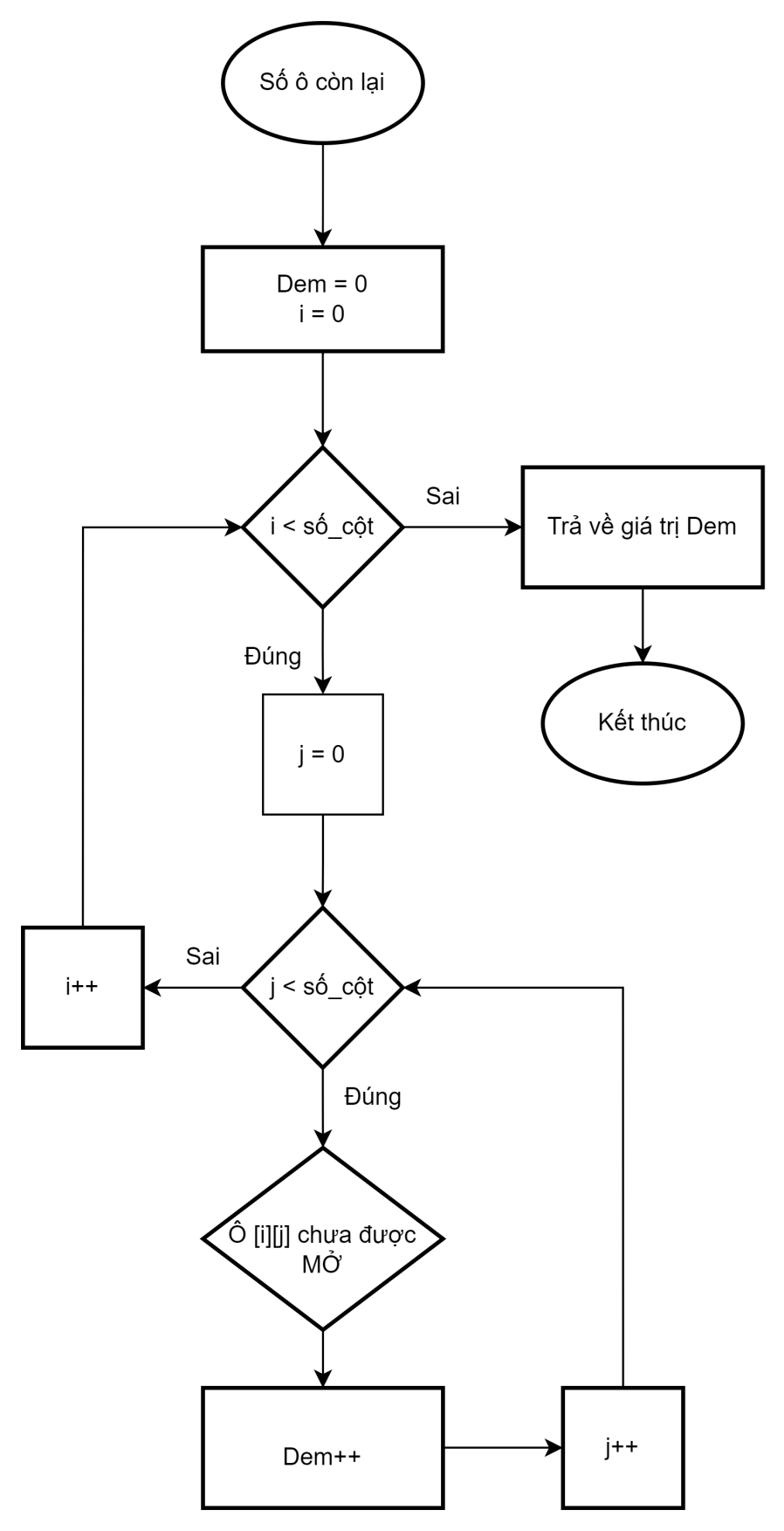


* Hàm đếm số lần chơi

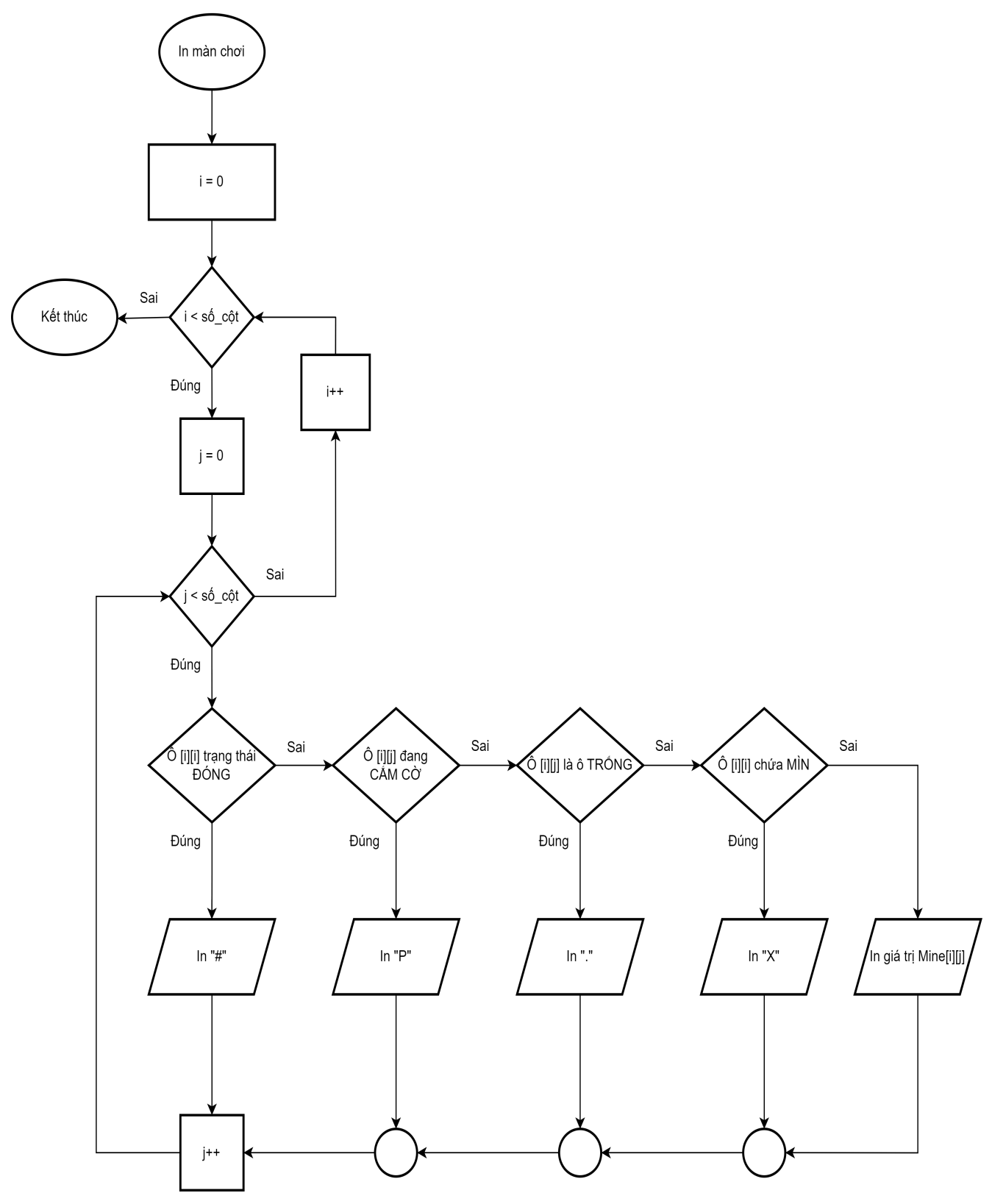
A black background with white squares and rectangles

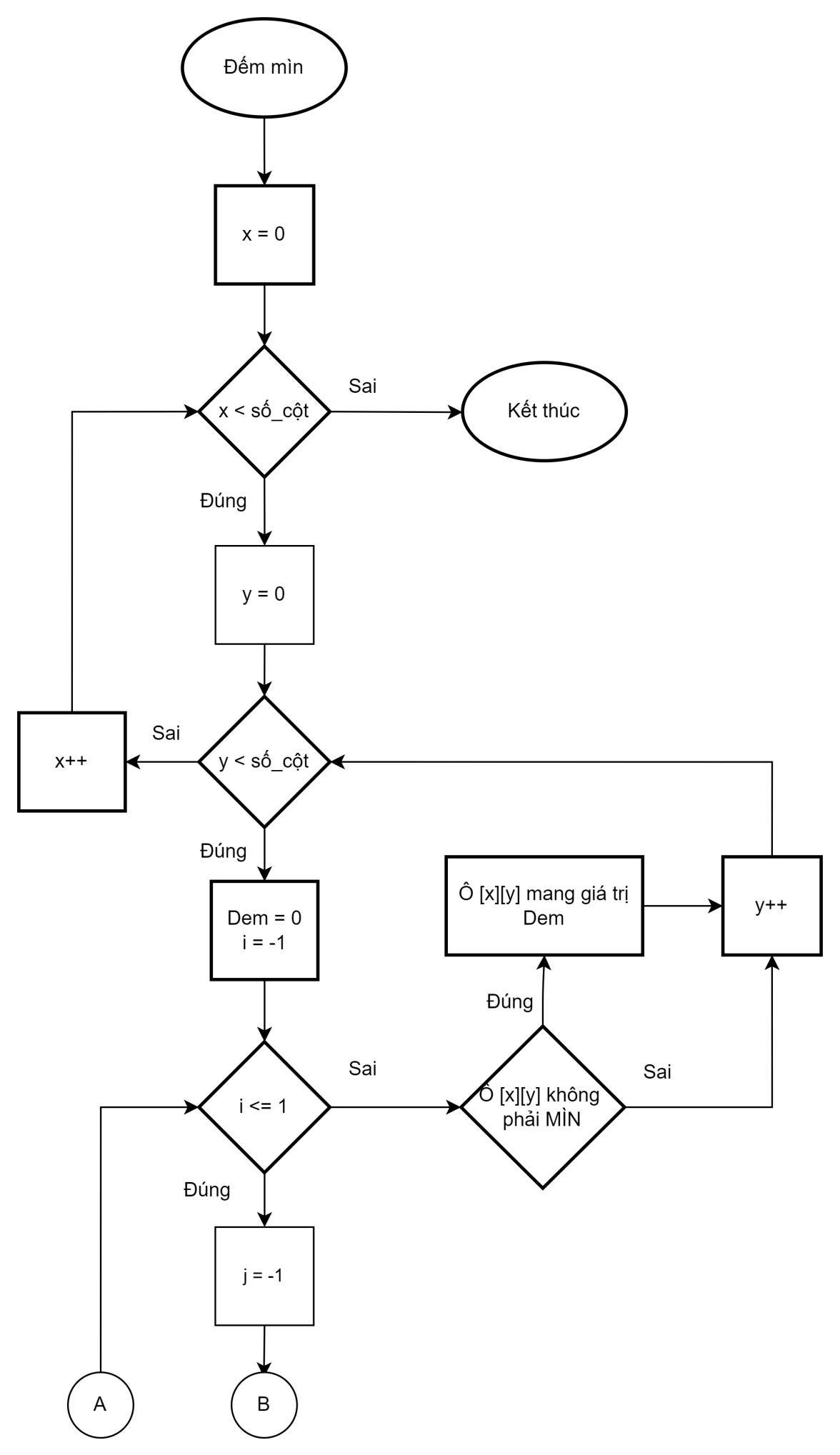
Description automatically generated

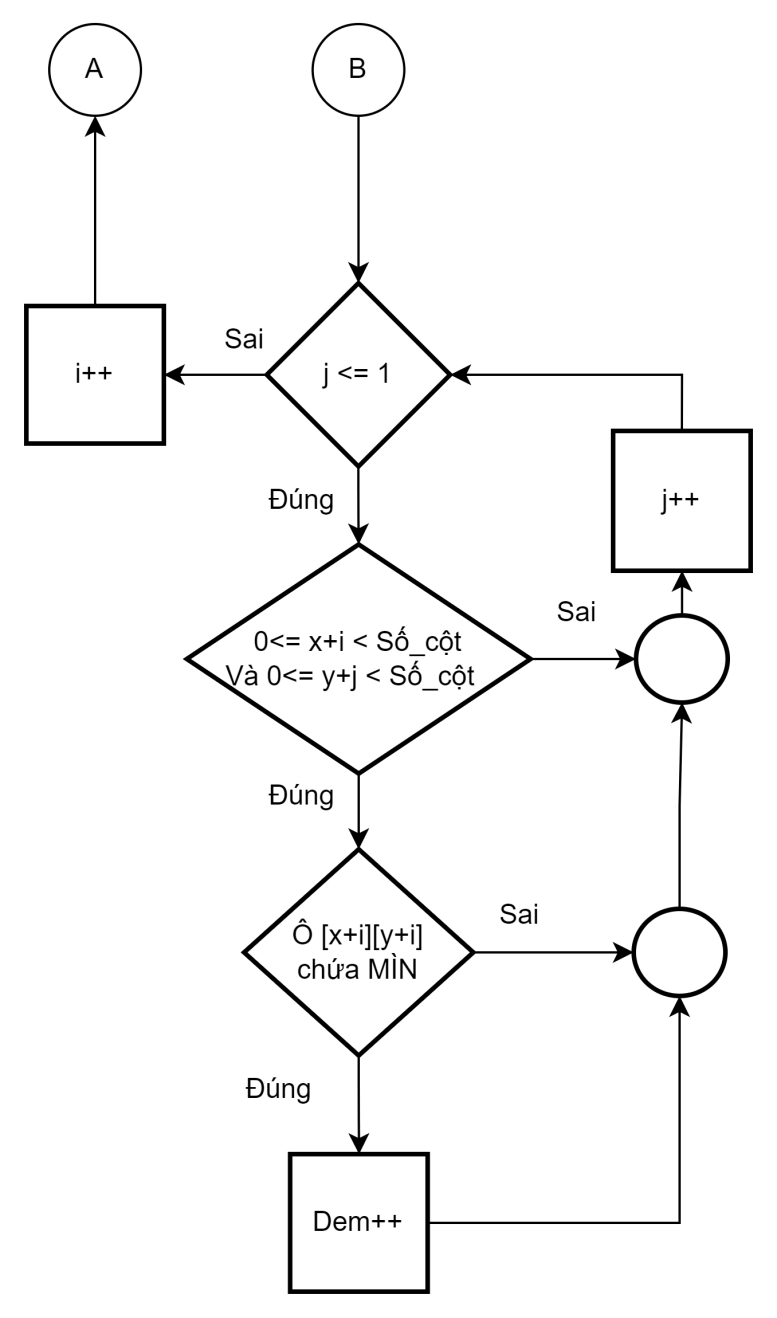
* Số ô còn lại



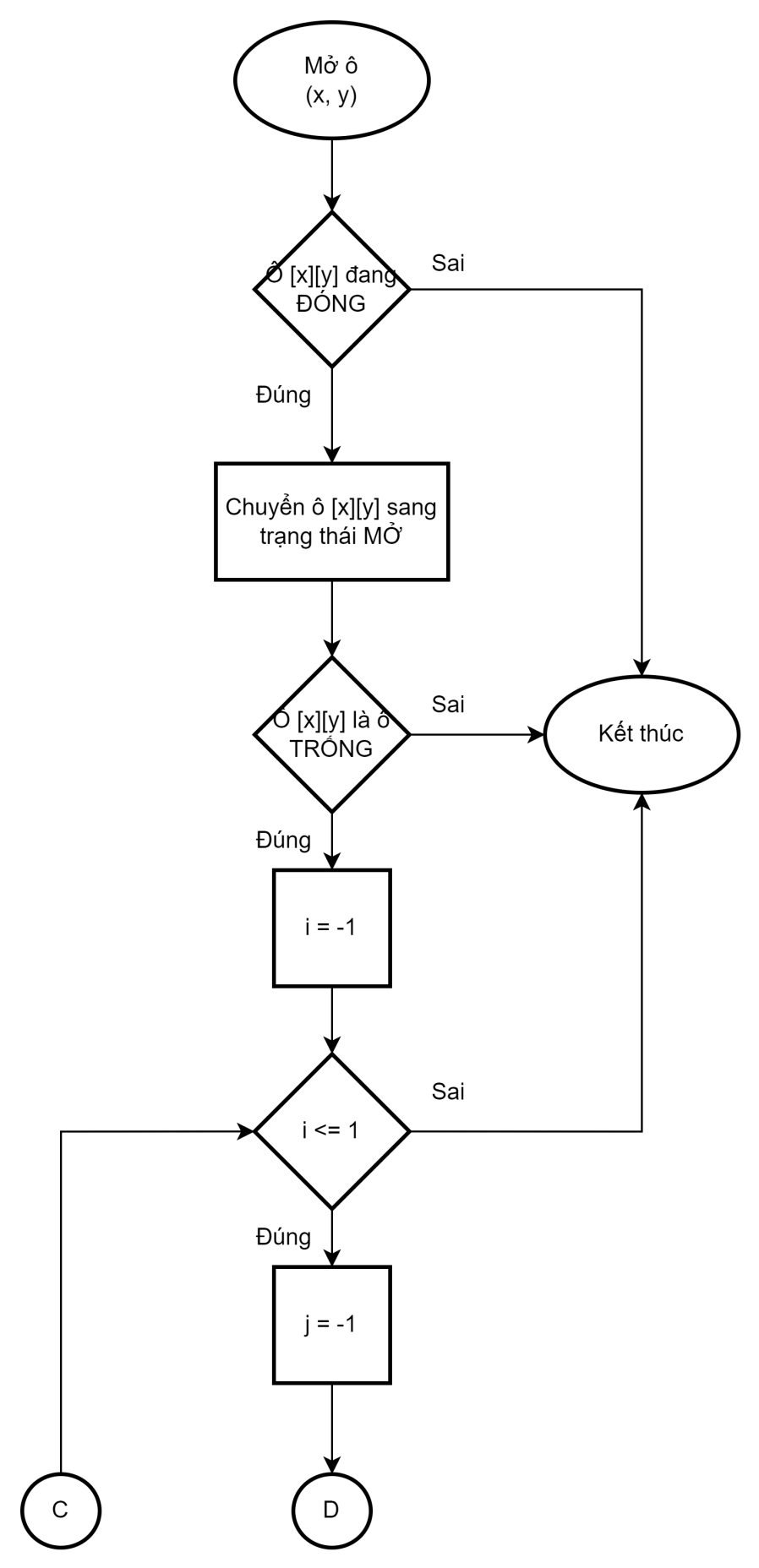
* Hàm in ra màn chơi



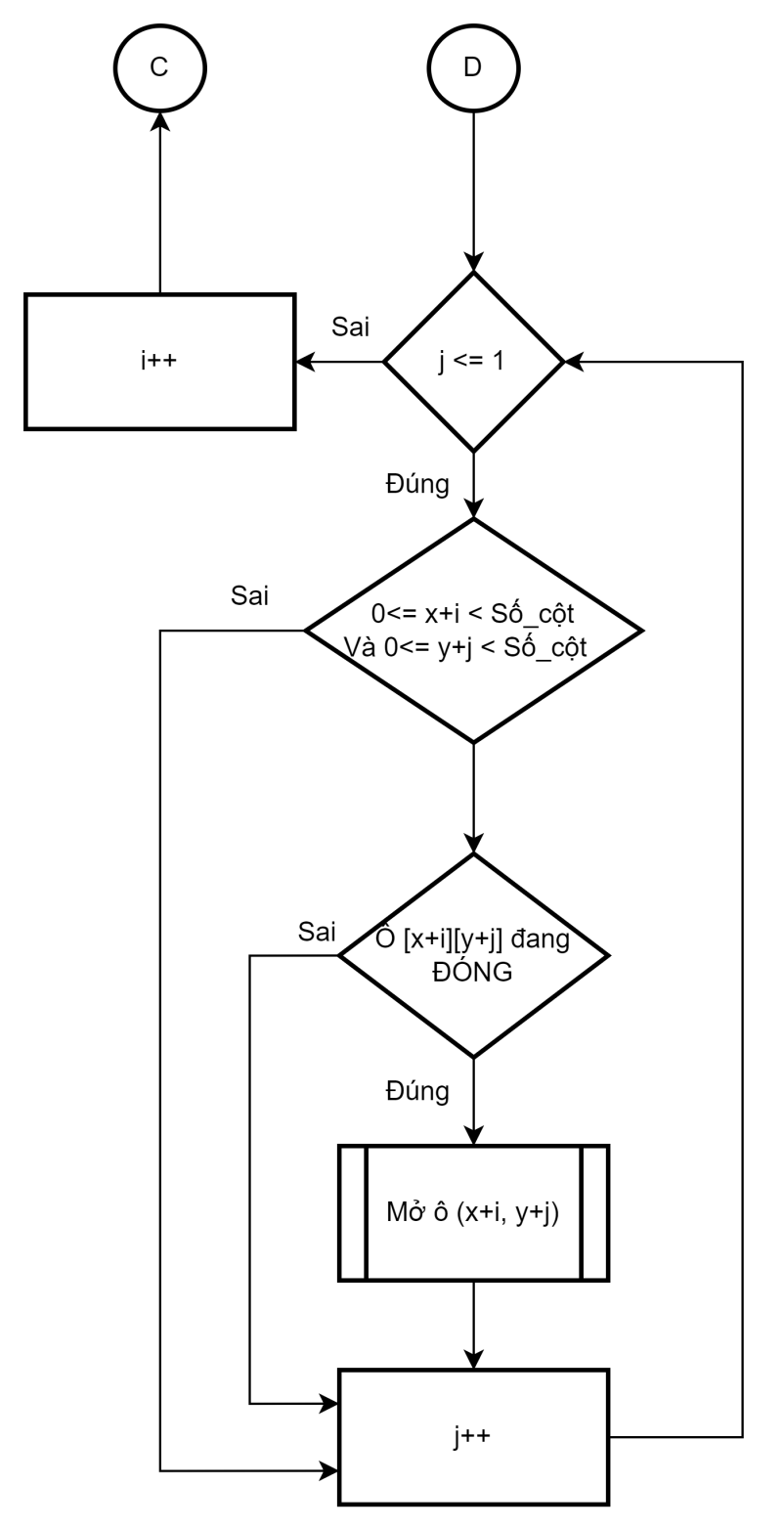
* Hàm đếm mìn 
* Hàm đếm mìn (tiếp theo)



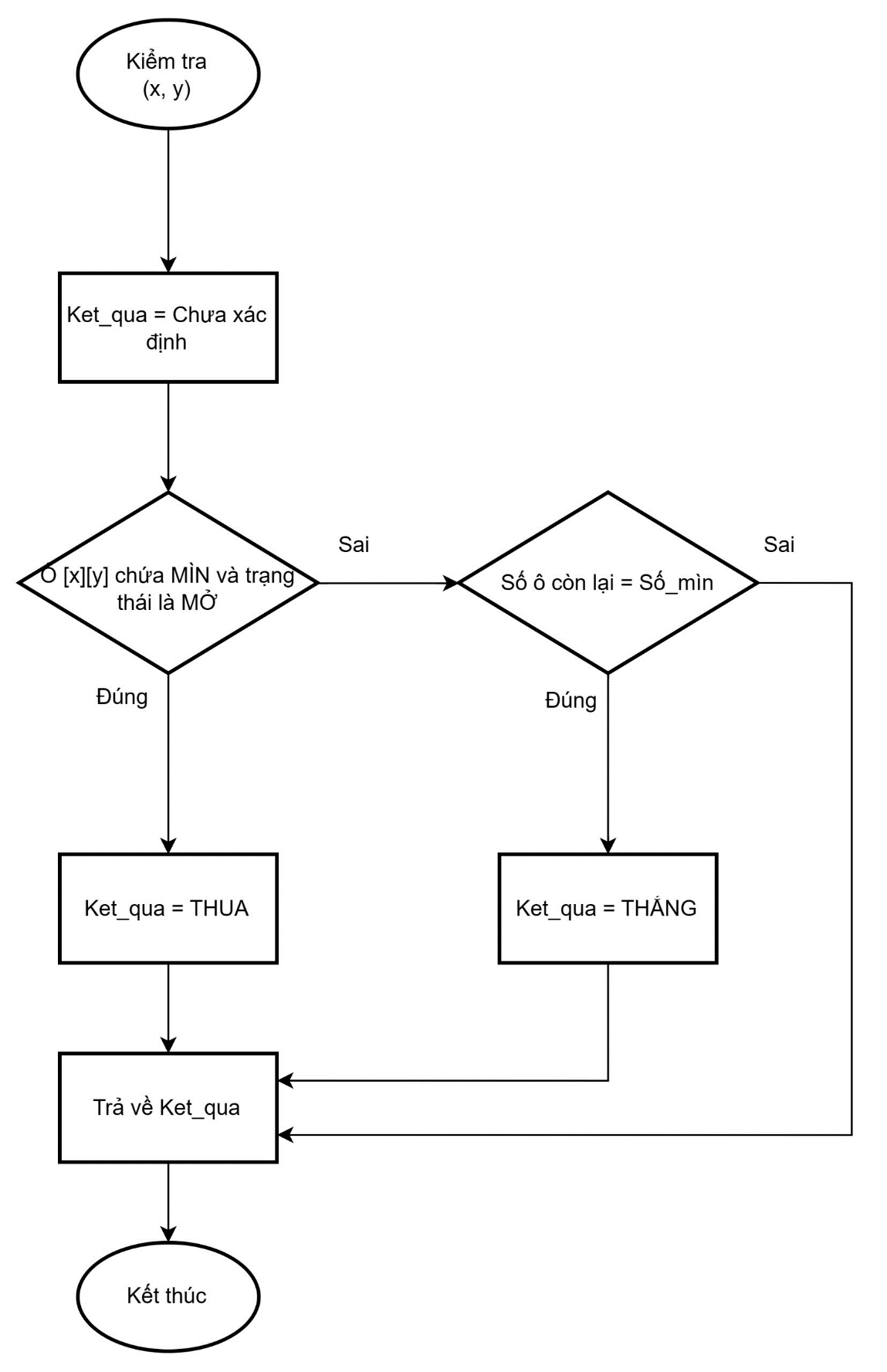
* Hàm mở ô



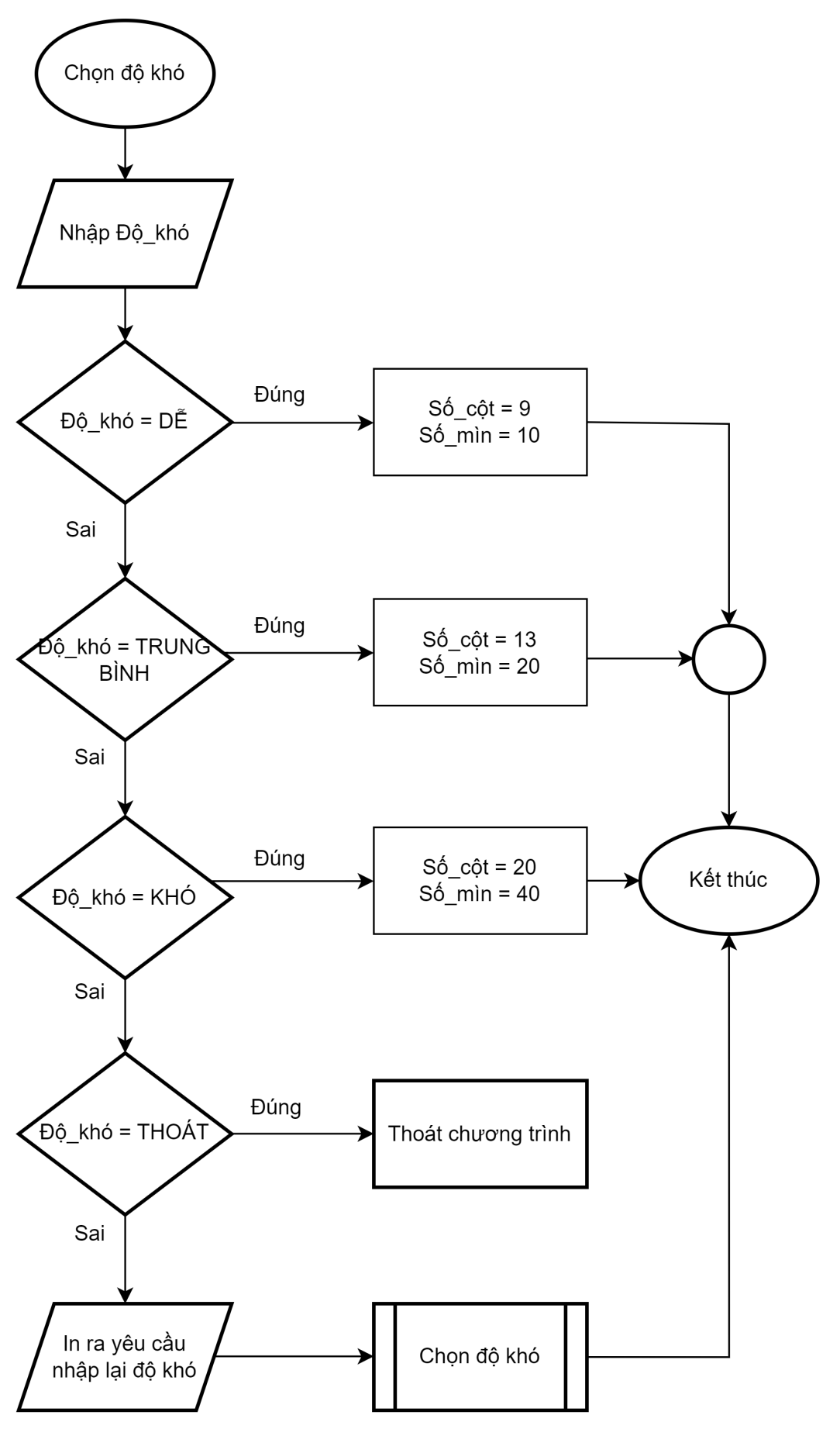
* Hàm mở ô (tiếp theo)



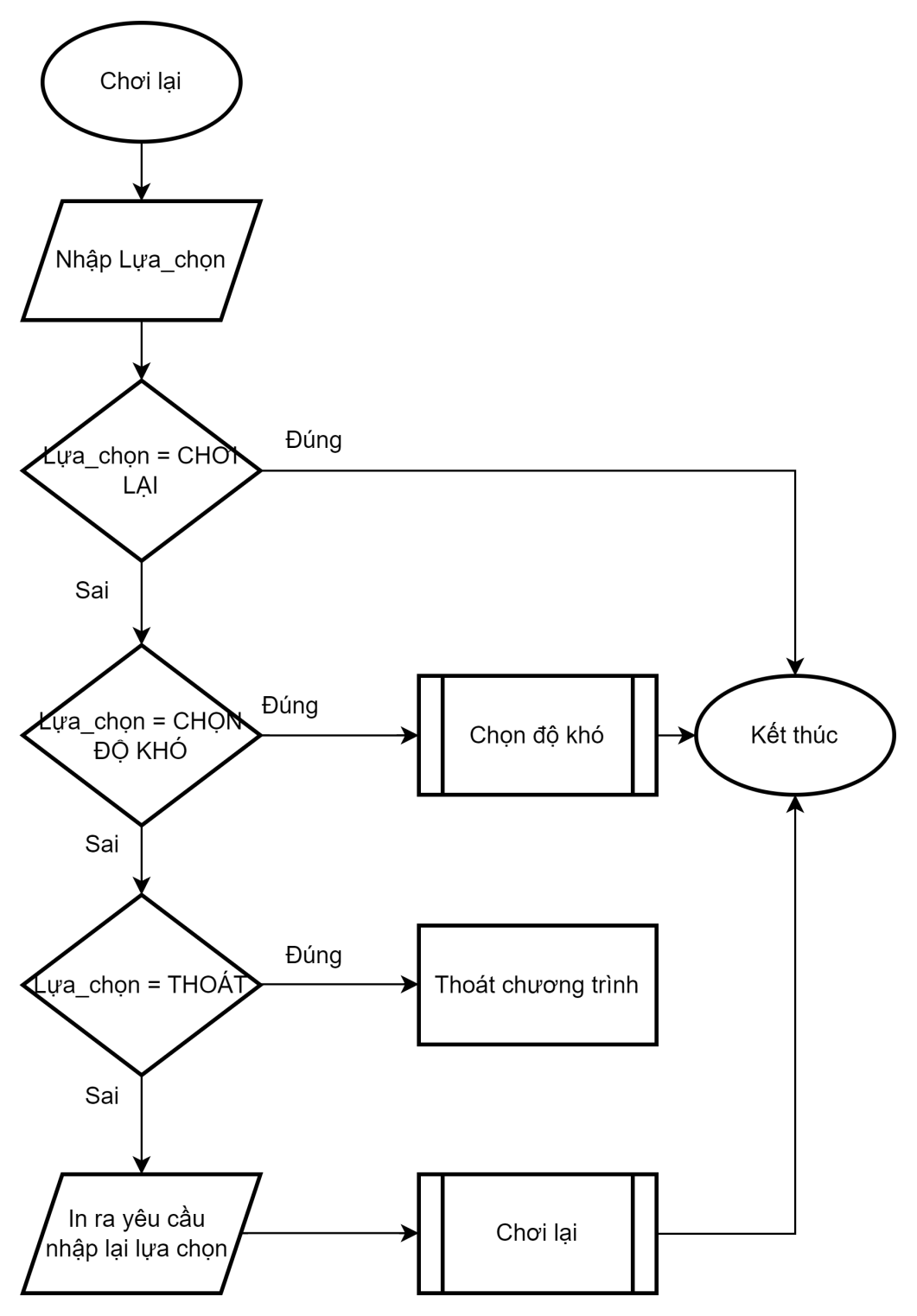
* Hàm kiểm tra



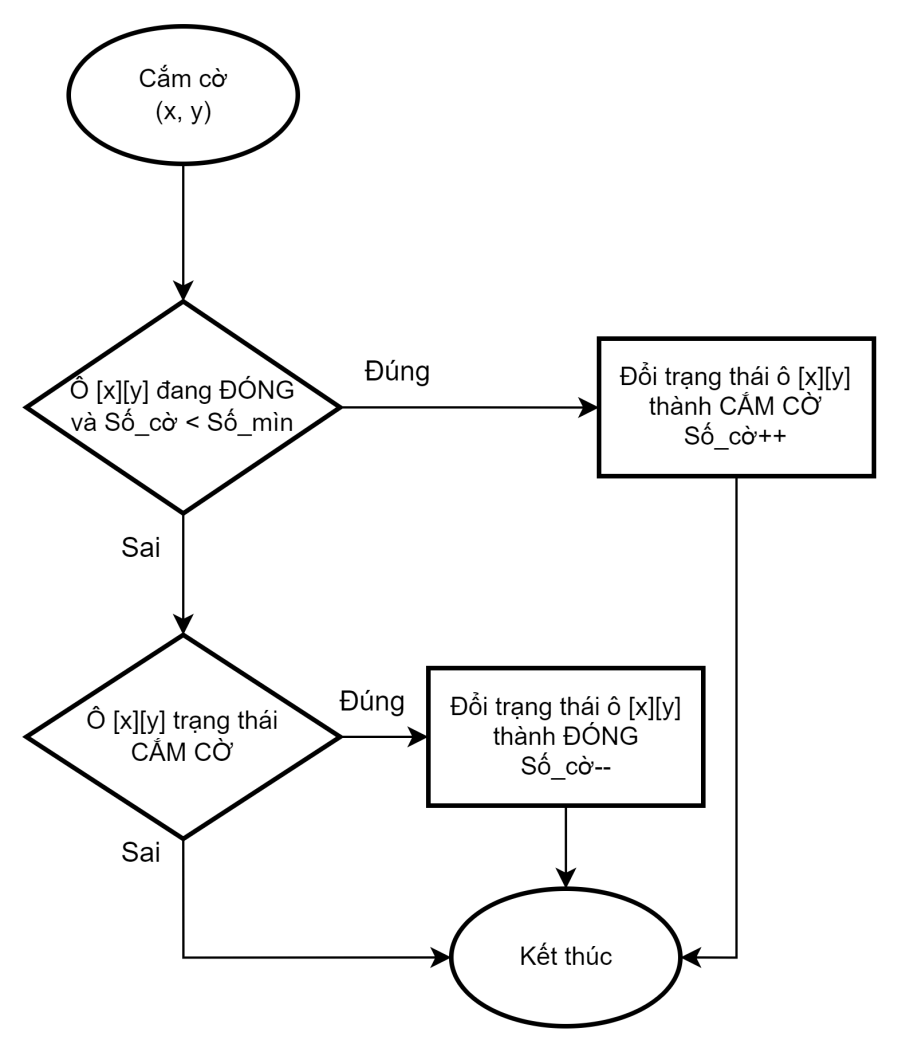
* Hàm chọn độ khó



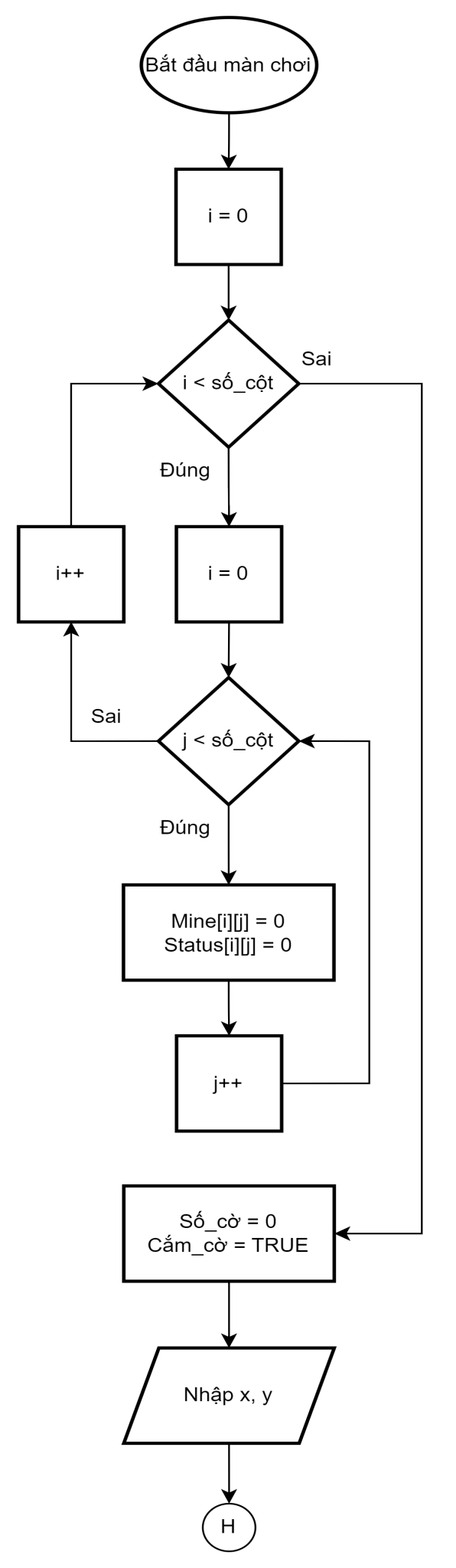
* Hàm chơi lại

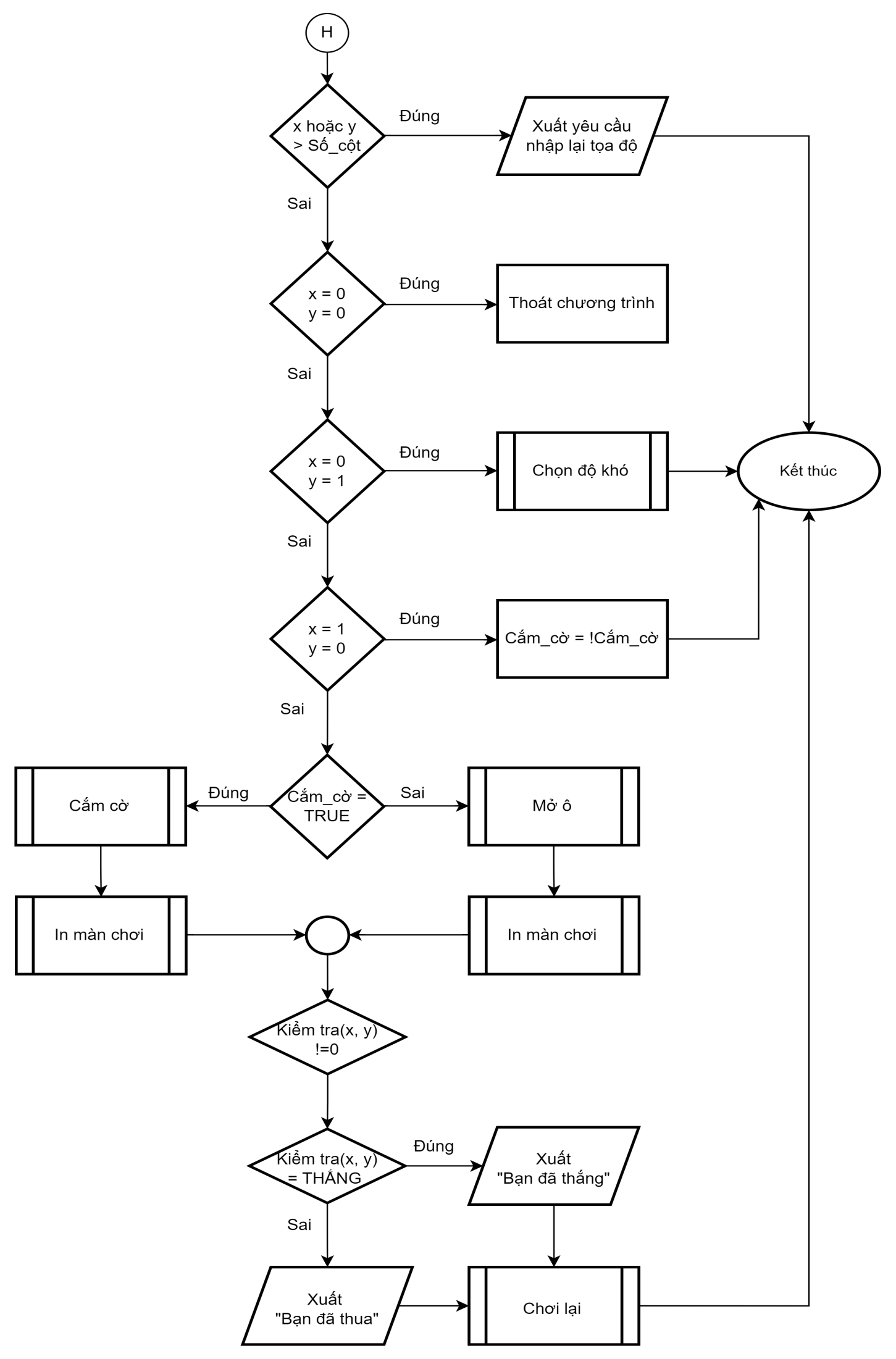


* Hàm cắm cờ



* Hàm bắt đầu màn chơi:





- Hàm bắt đầu chơi

(tiếp theo)

## Chương trình chi tiết

#include <stdio.h>

#include <stdlib.h>

#include <windows.h>

#include <time.h>

#define EASY '1'

#define MEDIUM '2'

#define HARD '3'

#define BLUE 1

#define GREEN 2

#define AQUA 3

#define RED 4

#define PURPLE 5

#define YELLOW 6

#define WHITE 7

#define GRAY 8

/\*ban do min\*/

#define MINE 9

#define EMPTY 0

int MineMap[20][20] = {0};

// ban do trang thai:

#define CLOSE 0

#define OPEN 1

#define FLAG 2

int StatusMap[20][20] = {0};

int So\_min, So\_cot, So\_co;

void PrintColor(const char text[], int color)

{

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hConsole, color); // set mau chu console thanh color

printf("%s", text);

SetConsoleTextAttribute(hConsole, FOREGROUND\_RED | FOREGROUND\_GREEN | FOREGROUND\_BLUE); // tra ve mau mac dinh

}

void demsolan()

{

int So\_lan = 0;

FILE \*file;

file = fopen("solan.txt", "r");

if (file != NULL)

fscanf(file, "%d", &So\_lan); // nhu ham scanf, no se gan gia tri doc tu file vao bien so

fclose(file); // dong file

So\_lan++; // cong them 1 sau moi lan chay

PrintColor("-------------------------------------------------------------------\n", GREEN | GRAY); // doi mau cua chu

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hConsole, GREEN | GRAY);

printf("------------------ SO LAN DA CHAY FILE LA %d LAN ------------------\n", So\_lan);

PrintColor("---------------------------------------------------------\n", GREEN | GRAY);

file = fopen("solan.txt", "w"); // mo file de ghi

if (file != NULL)

{

fprintf(file, "%d", So\_lan);

fclose(file);

}

}

void Tao\_min()

{

int i = 0, x, y;

srand(time(NULL)); // tao seed random

while (i < So\_min) // khi so min chua dat toi da thi tiep tuc them

{

x = rand() % So\_cot;

y = rand() % So\_cot;

if (MineMap[x][y] == 0) // neu vi tri x;y chua co min thi them min

{

MineMap[x][y] = 9;

i++;

}

}

}

void Dem\_min()

{

int count, r, c;

// xet tung o trong bang

for (c = 0; c < So\_cot; c++)

for (r = 0; r < So\_cot; r++)

{

// kiem tra lan luot 8 o xung quanh o dang xet

count = 0;

for (int i = -1; i <= 1; i++)

for (int j = -1; j <= 1; j++)

if ((r + i >= 0 && r + i < So\_cot) && (c + j >= 0 && c + j < So\_cot) && (i != 0 || j != 0)) // kiem tra cac o xung quanh co nam trong man choi khong

if (MineMap[r + i][c + j] == MINE) // neu o xung quanh o dang xet co min thi count+1

count++;

// neu o dang xet khong phai o min thì o do mang gia tri count

if (MineMap[r][c] != MINE)

MineMap[r][c] = count;

}

}

void Mo\_o(int r, int c)

{

int i, j;

if (StatusMap[r][c] == CLOSE)

{ // neu o da duoc mo hoac duoc cam co thi bo qua

StatusMap[r][c] = OPEN; // chuyen trang thai o dang xet thanh da mo

if (MineMap[r][c] == EMPTY) // neu o dang xet la o trong goi de quy mo 8 o xung quanh no

for (i = -1; i <= 1; i++)

for (j = -1; j <= 1; j++)

if ((r + i >= 0 && r + i < So\_cot) && (c + j >= 0 && c + j < So\_cot) && (i != 0 || j != 0))

if (StatusMap[r + i][c + j] == CLOSE)

Mo\_o(r + i, c + j);

}

}

void printMap()

{

// in so cot

printf(" ");

for (int t = 0; t < So\_cot; t++)

printf(" %d", (t + 1) % 10);

printf("\n");

// in so hang va trang thai o

for (int i = 0; i < So\_cot; i++)

{

for (int j = -1; j < So\_cot; j++)

{

if (j == -1) // in so hang

printf("%d ", (i + 1) % 10);

else if (StatusMap[i][j] == CLOSE) // neu o trang thai dong

PrintColor("# ", GREEN);

else if (StatusMap[i][j] == FLAG) // neu o trang thai cam co

PrintColor("P ", RED | GRAY);

else // neu o trang thai mo

{

switch (MineMap[i][j])

{

case EMPTY: // neu o trong thi in "."

PrintColor(". ", GRAY);

break;

case MINE:

PrintColor("X ", RED); // neu o chua min thì in "X"

break;

default:

MineMap[i][j] == 1 ? PrintColor("1 ", AQUA)

: MineMap[i][j] == 2 ? PrintColor("2 ", YELLOW)

: MineMap[i][j] == 3 ? PrintColor("3 ", PURPLE)

: MineMap[i][j] == 4 ? PrintColor("4 ", BLUE)

: printf("%d ", MineMap[i][j]); // in so min xung quanh o

break;

}

}

}

printf("\n");

}

}

/\*Tra ve so o con lai chua duoc mo\*/

int So\_o\_con\_lai()

{

int count = 0, i, j;

for (i = 0; i < So\_cot; i++)

for (j = 0; j < So\_cot; j++)

if (StatusMap[i][j] != OPEN)

count++;

return count;

}

/\*Tra ve ket qua thang thua cua nguoi choi

\*return 1: nguoi choi thang

\*return -1: nguoi choi thua

\*return 0: chua xac dinh\*/

int Kiem\_tra(int r, int c)

{

int win = 0;

if (MineMap[r][c] == MINE && StatusMap[r][c] == OPEN) // thua neu mo trung o co min

{

win = -1;

}

else if (So\_o\_con\_lai() == So\_min) // thang neu so o chua mo bang so min

{

win = 1;

}

return win;

}

void Cam\_co(int r, int c)

{

if (StatusMap[r][c] == CLOSE)

{

StatusMap[r][c] = FLAG;

So\_co++;

}

else if (StatusMap[r][c] == FLAG)

{

StatusMap[r][c] = CLOSE;

So\_co--;

}

}

void Chon\_level()

{

char level;

PrintColor("\n------------------ HAY CHON DO KHO CHO TRO CHOI ------------------\n\n", YELLOW | GRAY);

PrintColor("Nhap \"1\" de chon che do DE (9x9, 10 qua min) \n", PURPLE | GRAY);

PrintColor("Nhap \"2\" de chon che do TRUNG BINH (13x13, 20 qua min) \n", PURPLE | GRAY);

PrintColor("Nhap \"3\" de chon che do KHO (20x20, 30 qua min) \n", PURPLE | GRAY);

PrintColor("Nhap \"4\" de thoat khoi tro choi\n", PURPLE | GRAY);

fflush(stdin); // xoa bo nho dem de tranh bi lap

scanf("%c", &level);

switch (level)

{

case EASY:

So\_cot = 9;

So\_min = 5;

break;

case MEDIUM:

So\_cot = 13;

So\_min = 30;

break;

case HARD:

So\_cot = 20;

So\_min = 60;

break;

case '4':

exit(0);

break;

default:

system("cls");

PrintColor("Che do khong hop le, xin hay nhap lai!\n", RED | GRAY);

Chon\_level();

break;

}

system("cls");

}

void Choi\_lai()

{

char selection;

PrintColor("1.Choi lai \n", GREEN | GRAY);

PrintColor("2.Chon do kho\n", GREEN | GRAY);

PrintColor("3.Thoat \n", GREEN | GRAY);

while (1)

{

fflush(stdin); // xoa bo nho dem de tranh bi lap

scanf("%c", &selection);

switch (selection)

{

case '1':

system("cls");

return;

case '2':

system("cls");

Chon\_level();

return;

case '3':

exit(0);

break;

default:

PrintColor("Che do khong hop le, xin hay nhap lai!\n", RED | GRAY);

break;

}

}

system("cls");

}

void Newgame()

{

// reset mang ve 0

int i, j;

for (i = 0; i < So\_cot; i++)

for (j = 0; j < So\_cot; j++)

{

MineMap[i][j] = 0;

StatusMap[i][j] = 0;

}

So\_co = 0;

int Is\_flag\_on = 0; // trang thai cam co

int x, y;

Tao\_min();

Dem\_min();

printMap();

while (1)

{

PrintColor("-------------------------------------------------------------------\n", BLUE | GRAY);

PrintColor("Chon hang 0 va cot 0 de thoat khoi tro choi\n", BLUE | GRAY);

PrintColor("Chon hang 0 va cot 1 de chon lai che do\n", BLUE | GRAY);

PrintColor("Chon hang 1 va cot 0 de cam co, chon tuong tu de dung cam co\n", BLUE | GRAY);

PrintColor("-------------------------------------------------------------------\n", BLUE | GRAY);

PrintColor("Trang thai: ", PURPLE | GRAY);

Is\_flag\_on == 0 ? PrintColor("Mo o\n", YELLOW | GRAY) : PrintColor("Cam co\n", YELLOW | GRAY);

PrintColor("So co con lai: ", PURPLE | GRAY);

printf("%d\n", So\_min - So\_co);

PrintColor("Nhap toa do o\n", GREEN | GRAY);

PrintColor("Hang: ", GREEN | GRAY);

scanf("%d", &x);

PrintColor("Cot: ", GREEN | GRAY);

scanf("%d", &y);

if (x == 0 && y == 0) // thoat chuong trinh

exit(0);

if (x == 0 && y == 1) // thoat ham newgame de chon lai level

{

system("cls");

Chon\_level();

return;

}

if (x == 1 && y == 0)

{

Is\_flag\_on = !Is\_flag\_on; // dao trang thai cam co

system("cls");

printMap();

continue;

}

if (x > So\_cot || y > So\_cot) // bao loi neu x hoac y lon hon ban choi

{

system("cls");

printMap();

PrintColor("Toa do khong hop le, xin hay nhap lai!!!\n", RED);

continue;

}

x--; // dua x, y tu gia tri hang, cot ve chi so ma tran

y--;

if (Is\_flag\_on) // khi Is\_flag\_on = 1 (true)

{

Cam\_co(x, y);

system("cls");

printMap();

}

else

{

Mo\_o(x, y);

system("cls");

printMap();

}

if (Kiem\_tra(x, y) != 0)

{

system("cls");

if (Kiem\_tra(x, y) == -1)

PrintColor("\n================= BAN DA THUA =================\n", YELLOW | GRAY);

else

PrintColor("\n================= BAN DA CHIEN THANG =================\n", YELLOW | GRAY);

for (i = 0; i < So\_cot; i++) // mo tat ca o con lai

for (j = 0; j <= So\_cot; j++)

Mo\_o(i, j);

printMap();

Choi\_lai();

return;

}

}

}

int main()

{

system("cls");

demsolan();

Chon\_level();

while (1)

Newgame();

return 0;

}

# Kết luận

## Kết quả và hướng dẫn sử dụng

### Kết quả

* Trò chơi dò mìn đã hoàn thành các chức năng cơ bản như: tạo bản đồ mìn, mở ô, đếm số mìn xung quanh, cài đặt cấp độ khó, đặt cờ và kiểm tra kết quả thắng hoặc thua.
* Sử dụng thêm màu sắc để làm nổi bật các phần trong trò chơi.

### Hướng dẫn sử dụng trò chơi

* Khi bắt đầu trò chơi, người dùng được yêu cầu chọn một trong 3 cấp độ khó: Dễ, Trung bình hoặc Khó.
* Sau khi chọn cấp độ, bản đồ mìn sẽ được tạo ra và trò chơi bắt đầu.
* Sử dụng phím để chọn hàng và cột của ô bạn muốn mở.
* Khi mở một ô, nếu ô đó không có mìn, số trên ô sẽ hiển thị số mìn xung quanh ô. Nếu ô đó là mìn, trò chơi sẽ kết thúc và bạn sẽ thua.
* Bạn có thể đạt cờ trên ô mà bạn nghi ngờ là mìn bằng cách sử dụng chức năng “cắm cờ”. Sử dụng lại chức năng này để bỏ cờ.
* Bạn cần mở những ô không có mìn. Khi bạn mở tất cả các ô không có mìn, bạn sẽ thắng.
* Khi trò chơi kết thúc, bạn sẽ được thông báo về kết quả thắng hay thua và có thể tiếp tục chơi lại, chọn lại độ khó hoặc thoát khỏi trò chơi.

## Nhận xét (ưu, khuyết điểm, hướng phát triển...)

#### Ưu điểm

* Trò chơi dò mìn đơn giản, dễ sử dụng.
* Có sử dụng màu sắc để làm nổi bật các phần trong game, làm tăng trải nghiệm người chơi.
* Có hướng dẫn rõ ràng giúp người chơi dễ tương tác, điều chỉnh cài đặt trước khi bắt đầu trò chơi.

#### Khuyết điểm

* Chỉ hỗ trợ trên hệ điều hành Windows.
* Giao diện chưa được bắt mắt, chỉ là những màu sắc đơn giản.
* Chưa có các tính năng nâng cao như tính điểm, lưu điểm hoặc xếp hạng, tính toán thời gian chơi.
* Chưa có hướng dẫn cụ thể về cách chơi, quy tắt của trò chơi.

#### Hướng phát triển

* Cải thiện giao diện người dùng bằng cách sử dụng các thư viện đồ họa.
* Thêm tính năng tính điểm, lưu điểm cao và xếp hạng người chơi, ghi lại thời gian chơi.
* Thêm tính năng chọn ô bằng phím điều hướng hoặc sử dụng chuột để chọn.
* Tạo hướng dẫn với người dùng mới.
* Thêm các chế độ chơi mới.

# ****Tài liệu tham khảo****

1. Huong dan viet game Do min. URL: <https://yinyangit.wordpress.com/2011/01/23/c-h%C6%B0%E1%BB%9Bng-d%E1%BA%ABn-vi%E1%BA%BFt-game-do-min-minesweeper/>.
2. Giáo trình Lập trình căn bản