

BÁO CÁO PROJECT LỚN / KỸ THUẬT LẬP TRÌNH

Những sinh viên tham gia thực hiện : (Lớp: 010412410106 – CN2303CLCA)

Họ và tên:	MSSV:
Nguyễn Huy Lai	094205001839
Trần Minh Thiện	079205010726
Phạm Trương Anh Tú	051205000251

Ý tưởng của nhóm :

Tạo ra một game puzzle dạng 4x4 sử dụng giao diện text dễ nhìn , mục tiêu là đưa các số từ 1-15 về vị trí đúng để giành chiến thắng , trò chơi bao gồm phần Bắt đầu chơi - Hướng dẫn - Thoát .

Phần hướng dẫn sẽ hướng dẫn người chơi luật và cách chơi của game .

Khi chọn Bắt đầu chúng ta có thể tạo "người chơi mới" , đặt tên nhân vật hoặc sử dụng "người dùng" mà chúng ta đã tạo và lưu trước đó .

Trò chơi còn bao gồm cả các cấp độ khác nhau từ level 1-5 , mỗi level sẽ cung cấp một độ khó khác nhau để tạo thử thách cho người chơi , sau khi tạo nhân vật thì sẽ bắt đầu từ level 1 và phải hoàn thành câu đố để đạt đến những level cao hơn .

Phương án thực hiện :

Project trên được lập trình theo hướng đối tượng và hướng thủ tục .

1/ Class player :

Lớp này đại diện cho thông tin của mỗi người chơi với các thuộc tính như tên (name) và cấp độ (level).

Code :

```
// class Player {  
  
public:  
  
    char name[40];  
  
    int level;  
  
}; //
```

2/ Hàm Hiển Thị Menu (disp_o_menu) :

Hàm này được sử dụng để hiển thị các menu khác nhau trong trò chơi và trả về lựa chọn của người chơi.

Code :

```
// int disp_o_menu(char item[][25], int n, char* string);  
  
int disp_o_menu(Player* ptr, int n);  
  
int disp_o_menu(int level); //
```

3/ Hàm Chơi Trò Chơi (game_play):

Hàm này chứa logic chơi của trò chơi. Người chơi di chuyển ô để sắp xếp chúng và kiểm tra xem họ đã giành chiến thắng chưa.

Code :

```
// int game_play(int level[][4], int correct, int l); //
```

4/ Hàm Hiển Thị Bảng Trạng Thái (showconst):

Hàm này được sử dụng để hiển thị bảng trạng thái của trò chơi, như bảng trò chơi hiện tại và bảng mục tiêu.

Code :

```
// void showconst(int num[][4]); //
```

5/ Hàm Hiển Thị Menu Chính và Thông Tin Hướng Dẫn (main):

Hàm main chứa logic chính của chương trình, bao gồm hiển thị menu chính, hướng dẫn và bắt đầu trò chơi.

Code :

```
// int main(); //
```

6/ Sử Dụng Các Biến và Mảng Cho Các Cấp Độ Trò Chơi:

Sử dụng các biến và mảng để đại diện cho các cấp độ khác nhau của trò chơi và các thông tin liên quan.

Code :

```
// int level1[][4] = {{1,2,3,4},{5,6,7,8},{9,10,12,15},{13,14,11,99}};
```

```
int level2[][4] = {{1,6,2,3},{5,10,4,9},{8,14,7,12},{13,15,11,99}};
```

```
... ( những level khác ) //
```

7/ Sử Dụng Các Mảng Ký Tự Để Hiển Thị Thông Báo (char arrays):

Sử dụng các mảng ký tự để hiển thị các thông báo và tùy chọn trong trò chơi , giúp người dùng dễ dàng thao tác .

Code :

```
// char MMenu[][25] = {"Bắt đầu", "Hướng dẫn", "Thoát"};
```

```
char SMenu[][25] = {"Người mới", "Đang thoát"}; //
```

8/ Sử Dụng Các Lệnh và Hàm Điều Khiển Giao Diện Người Dùng (windows.h):

Sử dụng thư viện windows.h để điều khiển giao diện người dùng, chẳng hạn như màu sắc và kích thước cửa sổ console.

Code :

```
// #include <windows.h> //
```

Các kiến thức được sử dụng trong project :

- Ngôn Ngữ Lập Trình C++:

Sử dụng để triển khai toàn bộ dự án, từ khai báo biến đến xử lý logic chính.

- Lập Trình Hướng Thủ Tục và Hướng Đối Tượng (Partial):

Sử dụng lập trình hướng thủ tục cho nhiều phần của dự án, đồng thời có một số yếu tố của lập trình hướng đối tượng (OOP) như lớp Player.

- Mảng Đa Chiều:

Sử dụng mảng đa chiều để biểu diễn bảng trò chơi và các cấp độ khác nhau.

- Đọc/Ghi Dữ Liệu Từ/Tới Tập (fstream):

Sử dụng để lưu trữ thông tin về người chơi vào tệp "data.txt".

- Giao Diện Người Dùng Cơ Bản (Console):

Sử dụng hàm và lệnh từ thư viện windows.h để điều khiển giao diện người dùng trong môi trường console.

- Xử Lý Sự Kiện Người Dùng:

Sử dụng hàm GetAsyncKeyState để theo dõi và xử lý sự kiện từ bàn phím người dùng.

- Thư Viện Thời Gian và Ngủ (<ctime>, Sleep()):

Sử dụng để sinh số ngẫu nhiên và tạo hiệu ứng ngủ trong trò chơi.

- Sử Dụng Chuỗi Ký Tự (char Arrays):

Sử dụng chuỗi ký tự để biểu diễn tên người chơi và các thông báo trong trò chơi.

- Thư Viện Hiệu Ứng Đồ Họa Cơ Bản (Màu Sắc Console):

Sử dụng hàm SetConsoleTextAttribute để đặt màu sắc cho văn bản trong console.

- Sử Dụng Hàm Beep():

Sử dụng để tạo âm thanh khi người chơi giành chiến thắng.

Từ những kiến thức trên nhóm em bắt đầu viết project như sau :

Project có 2 đối tượng chính :

1/ Mảng và Biến Cấp Độ Trò Chơi:

```
// int level1[][4] = {{1,2,3,4},{5,6,7,8},{9,10,12,15},{13,14,11,99}};
```

```
int level2[][4] = {{1,6,2,3},{5,10,4,9},{8,14,7,12},{13,15,11,99}};
```

Các mảng cấp độ khác... //

2/ Class player :

```
// class Player {
```

```
public:
```

```
    char name[40];
```

```
int level;  
}; //
```

Các quan hệ trong project :

1/ Quan Hệ Với Cấp Độ Trò Chơi:

Các mảng level1, level2, ..., level5 đại diện cho các cấp độ khác nhau của trò chơi. Mỗi mảng này chứa một bảng số nguyên 4x4 biểu diễn trạng thái ban đầu của trò chơi ở mỗi cấp độ.

Mảng correct chứa số lượng ô được đặt đúng cho mỗi cấp độ. Đây là thông tin để kiểm tra xem người chơi đã giành chiến thắng ở mỗi cấp độ hay chưa.

2/ Quan Hệ Với Người Chơi (Lớp Player):

Lớp Player đại diện cho thông tin của mỗi người chơi. Mỗi đối tượng của lớp này bao gồm tên (name) và cấp độ (level) của người chơi.

Mảng data là một mảng của các đối tượng Player, nơi lưu trữ thông tin về người chơi. Mỗi phần tử của mảng này đại diện cho một người chơi cụ thể.

Mảng con trỏ ptr chứa con trỏ đến các đối tượng Player, có thể được sử dụng để quản lý và chọn người chơi trong quá trình chơi.

3/ Quan Hệ Với Tập Tin (fstream):

Sử dụng fstream để đọc và ghi thông tin người chơi vào tệp "data.txt". Thông tin này bao gồm tên và cấp độ của người chơi.

4/ Giao Diện Người Dùng (Console):

Sử dụng các lệnh từ thư viện windows.h để tương tác với người chơi trong môi trường console.

5/ Xử Lý Sự Kiện Người Dùng:

Sử dụng hàm GetAsyncKeyState để theo dõi và xử lý sự kiện từ bàn phím người chơi.

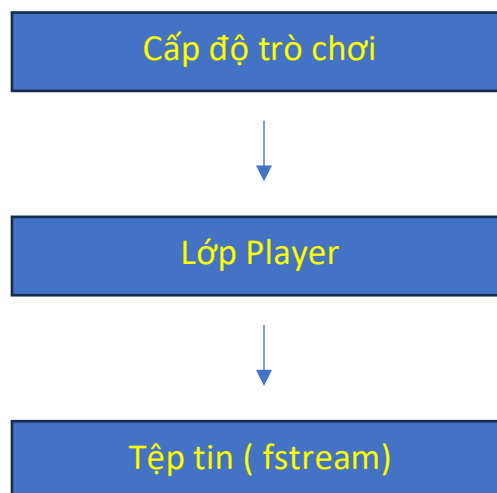
6/ Các Hàm Thực Hiện Trò Chơi (game_play):

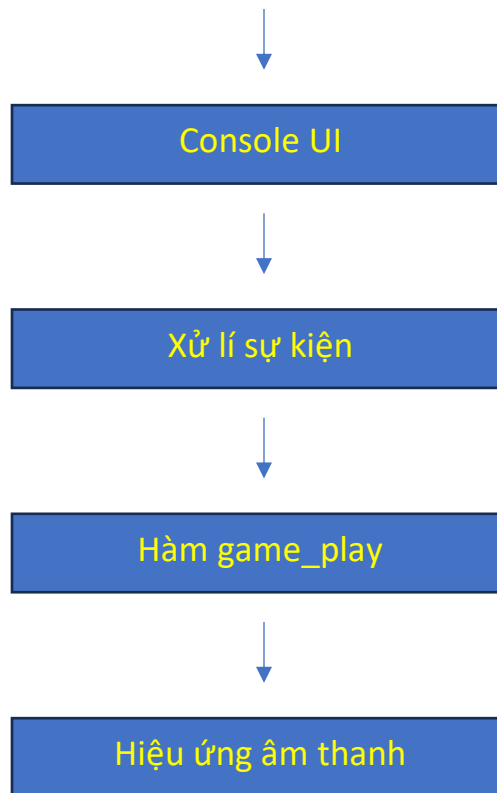
Hàm game_play thực hiện logic chính của trò chơi, bao gồm việc di chuyển các ô trong bảng và kiểm tra xem người chơi đã giành chiến thắng hay chưa.

7/ Hiệu Ứng Âm Thanh (Beep):

Sử dụng hàm Beep() để tạo hiệu ứng âm thanh khi người chơi giành chiến thắng.

SƠ ĐỒ MINH HỌA :





Project và giải thích :

```
#include<iostream>
```

```
#include<windows.h> // Thêm thư viện windows.h để sử dụng các hàm cụ thể của Windows
```

```
#include<iomanip> // Thêm thư viện iomanip để sử dụng các trình điều khiển để định dạng đầu ra
```

```
#include<fstream> // Thêm thư viện fstream để thực hiện các thao tác với tệp
```

```
class Player{ // Định nghĩa một lớp có tên là Player
```

```
public:
```



```
char name[40]; // Khai báo một mảng ký tự công khai có tên là name với  
kích thước là 40
```

```
int level; // Khai báo một biến số nguyên công khai có tên là level  
};
```

```
using namespace std; // Sử dụng không gian tên chuẩn
```

```
int board[][4]={{1,2,3,4},{5,6,7,8},{9,10,11,12},{13,14,15,99}}; // Khai báo và  
khởi tạo mảng hai chiều
```

```
int main() // Hàm main - điểm bắt đầu của chương trình
```

```
{  
    SetConsoleOutputCP(CP_UTF8); // Đặt mã hóa đầu ra của console là UTF-  
8
```

```
// Khai báo các hàm
```

```
void showconst(int num[][4]);
```

```
int game_play(int level[][4],int correct,int l);
```

```
int disp_o_menu(char item[][25],int n,char* string);
```

```
int disp_o_menu(int level);
```

```
Player* disp_o_menu(Player* ptr,int n);
```

```
// Khai báo và khởi tạo các biến và mảng
```

```
Player data[40],*ptr[40];
```

```
char MMenu[][25]={"Bắt đầu","Hướng dẫn","Thoát"},name[40];
```

```

char SMenu[][25]={"Người mới","Đang thoát"};
char yes_no[][25]={"Có","Không"};
int level1[][4]={{1,2,3,4},{5,6,7,8},{9,10,12,15},{13,14,11,99}}; // level 1
int level2[][4]={{1,6,2,3},{5,10,4,9},{8,14,7,12},{13,15,11,99}}; // level 2
int level3[][4]={{1,6,3,5},{4,6,2,8},{13,10,11,12},{9,14,15,99}}; // level 3
int level4[][4]={{1,7,3,6},{5,4,2,8},{9,13,11,12},{10,14,15,99}}; // level 4
int level5[][4]={{1,4,3,7},{5,6,2,8},{9,10,13,12},{11,14,15,99}}; // level 5
int correct[5]={2,13,4,4,5};

// Đánh dấu điểm bắt đầu của menu
menu:

int choice1=disp_o_menu(MMenu,3,"Main Menu"); // Hiển thị menu
chính và lấy lựa chọn của người dùng

if(choice1==0){ // Nếu người dùng chọn "Bắt đầu"
    Sleep(100); // Dừng chương trình trong 100 mili giây
    int choice2=disp_o_menu(SMenu,2,"Người dùng"); // Hiển thị menu
    người dùng và lấy lựa chọn của người dùng
    fstream file; // Khai báo một đối tượng file
    char name[40];int level,levels;
    file.open("data.txt",ios::app|ios::in|ios::out); // Mở file "data.txt" để đọc
    và ghi
    int id=0;
    while(file>>name>>level) // Đọc dữ liệu từ file
    {

```

```

strcpy(data[id].name,name); // Sao chép tên vào mảng data
data[id].level=level; // Gán level cho phần tử thứ id trong mảng data
ptr[id]=&data[id]; // Gán địa chỉ của phần tử thứ id trong mảng data
cho ptr[id]

    id++;
}

file.close(); // Đóng file

if(choice2==0) // Nếu người dùng là người mới
{
    int out=1;
    do{
        system("cls"); // Xóa màn hình console

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),9); // Đặt
màu chữ của console
        cout<<"Nhập tên mà bạn muốn nhập. ";
        cin>>name;
        out=1;
        for(int j=0;j<id;j++)
        {
            if(!strcmp(data[j].name,name)) // So sánh tên vừa nhập với các
tên đã có
            {
                out=0;
                cout<<"Tên của bạn đã tồn tại!"<<endl;
                Sleep(900); // Dừng chương trình trong 900 mili giây
                break;
            }
        }
    }
}

```

```

}while(out!=1);
while(GetAsyncKeyState(VK_RETURN)!=0)
    cout<<".....";
while(GetAsyncKeyState(VK_ESCAPE)!=0);
level=1;
levels=level;
}
else // Nếu người dùng không phải là người mới
{
    cout<<"Đang tải menu..";
    cout<<".";Sleep(200);
    cout<<".";Sleep(200);
    Player* select=disp_o_menu(ptr[0],id); // Hiển thị menu người dùng
    và lấy lựa chọn của người dùng
    strcpy(name,select->name); // Sao chép tên đã chọn vào biến name
    system("cls");
    cout<<"đang xử lý...";
    cout<<".";Sleep(200);
    level=select->level; // Gán level cho biến level
    levels=disp_o_menu(level); // Hiển thị menu level và lấy lựa chọn của
    người dùng
    Sleep(500);
}

system("cls");
int ret;
do{
    system("cls");
    char a;
    switch(levels) // Chọn level dựa trên lựa chọn của người dùng
    {
        case 1:

```

```

        ret=game_play(level1,correct[0],1);
        break;
    case 2:
        ret=game_play(level2,correct[1],2);
        break;
    case 3:
        ret=game_play(level3,correct[2],3);
        break;
    case 4:
        ret=game_play(level4,correct[3],4);
        break;
    case 5:
        ret=game_play(level5,correct[4],5);
        break;
    }
    // Kiểm tra nếu ret bằng 1000
    if(ret==1000)
    {
        // Xóa màn hình console
        system("cls");

        // In ra thông báo cho người chơi
        cout<<"you ";Sleep(250);
        cout<<"wished ";Sleep(250);
        cout<<"to ";Sleep(250);
        cout<<"escape ";Sleep(250);

        // Chờ người dùng nhấn phím Enter

```

```
while(GetAsyncKeyState(VK_RETURN)!=0)
```

```
    cout<<".....";
```

```
    // Quay lại menu
```

```
    goto menu;
```

```
}
```

```
// Tăng levels lên 1
```

```
levels++;
```

```
// Kiểm tra nếu levels lớn hơn level và levels khác 6
```

```
if(levels>level&&levels!=6)
```

```
{
```

```
    // Đặt level bằng levels
```

```
    level=levels;
```

```
    // Hiển thị menu cho người chơi và lưu kết quả vào biến c
```

```
    int c=disp_o_menu(yes_no,2,"Bạn có muốn lưu tiến trình trò chơi lại ?");
```

```
    // Kiểm tra nếu c bằng 0
```

```
    if(c==0)
```

```
{
```

```
    // In ra thông báo đang lưu
```

```
    cout<<"Đang lưu.";
```

```
// Khởi tạo biến x bằng id
```

```
int x=id;
```

```
cout<<".";
```

```
// Mở file data.txt để ghi
```

```
file.open("data.txt",ios::out);
```

```
cout<<".";
```

```
// Đóng file
```

```
file.close();
```

```
cout<<".";
```

```
// Mở file data.txt để ghi tiếp
```

```
file.open("data.txt",ios::out|ios::app);
```

```
cout<<".";
```

```
// Duyệt qua mảng data
```

```
while((x-1)>=0){
```

```
    // Kiểm tra nếu tên trong data trùng với tên người chơi
```

```
    if(!strcmp(data[x-1].name,name))
```

```
    {
```

```
        // Đặt level của người chơi trong data bằng level hiện tại
```

```
        data[x-1].level=level;
```

```

    }

    // Ghi tên và level của người chơi vào file
    file<<data[x-1].name<<endl<<data[x-1].level<<endl;
    x--;
}

// Kiểm tra nếu choice2 bằng 0
if(choice2==0)
{
    // Ghi tên và level của người chơi vào file
    file<<name<<endl<<level<<endl;
}

cout<<".";

// Đóng file
file.close();
}
else{
    // Chờ người dùng nhấn phím Enter
    while(GetAsyncKeyState(VK_RETURN)!=0);
}
}

```



```
// Kiểm tra nếu levels lớn hơn 5
```

```
if(levels>5)
```

```
{
```

```
    // In ra thông báo hoàn thành game
```

```
    cout<<endl<<"Game completed";
```

```
    // Tạm dừng chương trình trong 40 mili giây
```

```
    Sleep(40);
```

```
    // Thoát khỏi vòng lặp
```

```
    break;
```

```
}
```

```
// Chờ người dùng nhấn phím Enter
```

```
while(GetAsyncKeyState(VK_RETURN)!=0);
```

```
// Hiển thị menu cho người chơi và lưu kết quả vào biến c
```

```
int c=disp_o_menu(yes_no,2,"Bạn có muốn chơi cấp độ tiếp theo?");
```

```
// Kiểm tra nếu c bằng 1
```

```
if(c==1)
```

```
{
```

```
    // Chờ người dùng nhấn phím Enter
```

```
while(GetAsyncKeyState(VK_RETURN)!=0);

// Quay lại menu
goto menu;
}
}while(1);
}

else if(choice1==1){
    // Xóa màn hình console
    system("cls");

    // Đặt màu cho console
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),2);

    // In ra thông báo cho người chơi
    cout<< "-----Hướng dẫn cách chơi-----"<<endl<<endl;

    // Đặt màu cho console
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),3);

    // In ra các thông báo cho người chơi
    cout<< "-----Ở đây sẽ có ngẫu nhiên số từ 1-15."<<endl;
    cout<< "-----Nhiệm vụ của bạn là làm sao để nó về đúng vị trí của nó."<<endl;
    cout<< "-----Ví dụ:"<<endl;

    // Hiển thị mảng level1
```

```
showconst(level1);
```

```
// Đặt màu cho console
```

```
SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),3);
```

```
cout<<endl<<"-----trở thành..."<<endl<<endl;
```

```
// Hiển thị mảng board
```

```
showconst(board);
```

```
// Tạm dừng chương trình
```

```
system("pause");
```

```
// In ra các thông báo cho người chơi
```

```
cout<<"-----Chỉ duy nhất ô trống là bạn có thể điều khiển qua lại với  
nhau."<<endl;
```

```
cout<<"-----Hãy sử dụng phím mũi tên <^> để di chuyển ô trống  
đó"<<endl;
```

```
cout<<"                Good luck for you!!!                "<<endl;
```

```
// Chờ người dùng nhấn phím Enter
```

```
while(GetAsyncKeyState(VK_RETURN)!=0);
```

// Quay lại menu

goto menu;

}

else

return 0;

}

// Hàm hiển thị mảng 2 chiều num

void showconst(int num[][4]){

 // Đặt màu cho console

 SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),6);

 // Duyệt qua mảng 2 chiều

 for(int i=0;i<4;i++){

 for(int j=0;j<4;j++){

 // Nếu phần tử là 99, in ra "__"

 if(num[i][j]==99)

 cout<<setw(3)<<"__";

 else

 // Ngược lại, in ra giá trị của phần tử

 cout<<setw(3)<<num[i][j];

 }

 cout<<endl;

}

```
// Đặt lại màu cho console
```

```
SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),4);
```

```
}
```

```
// Hàm hiển thị menu người chơi
```

```
Player* disp_o_menu(Player* ptr,int n){
```

```
    int pointer=0;
```

```
    for(;;){
```

```
        Player* ptrt=ptr;
```

```
        system("cls");
```

```
        // Đặt màu cho console
```

```
        SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),2);
```

```
        cout<<setw(40)<<"Name"<<setw(5)<<" Level"<<endl;
```

```
        // Đặt lại màu cho console
```

```
        SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),15);
```

```
        for(int i=0;i<n;i++)
```

```
        {
```

```
            if(i==pointer)
```

```
                // Đặt màu cho console
```

```
                SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),11);
```

```
                cout<<setw(40)<<ptrt->name<<" "<<setw(5)<<ptrt->level<<endl;
```

```
                // Đặt lại màu cho console
```

```
                SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),15);
```

```
ptrt++;  
}  
  
// Xử lý sự kiện từ bàn phím  
while(true)  
{  
    if(GetAsyncKeyState(VK_UP)!=0)  
    {  
        pointer-=1;  
        if(pointer== -1)  
            pointer=n-1;  
        break;  
    }  
    else if(GetAsyncKeyState(VK_DOWN)!=0)  
    {  
        pointer+=1;  
        if(pointer==n)  
            pointer=0;  
        break;  
    }  
    else if(GetAsyncKeyState(VK_RETURN)!=0)  
        return ptr+pointer;  
}  
  
Sleep(120);
```

```
}  
}
```

// Hàm hiển thị menu với các mục được lưu trong mảng item

```
int disp_o_menu(char item[][25],int n,char* string){  
    int pointer=0;  
    for(;;){  
        system("cls");  
        // Đặt màu cho console  
        SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),2);  
        cout<<string<<endl<<endl;  
        // Đặt lại màu cho console  
        SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),15);  
        for(int i=0;i<n;i++)  
        {  
            if(i==pointer)  
                // Đặt màu cho console  
                SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),11);  
            cout<<item[i]<<endl;  
            // Đặt lại màu cho console  
            SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),15);  
        }  
    }  
}
```

// Xử lý sự kiện từ bàn phím

while(true)

{

if(GetAsyncKeyState(VK_UP)!=0)

{

pointer-=1;

if(pointer== -1)

pointer=n-1;

break;

}

else if(GetAsyncKeyState(VK_DOWN)!=0)

{

pointer+=1;

if(pointer==n)

pointer=0;

break;

}

else if(GetAsyncKeyState(VK_RETURN)!=0)

return pointer;

}

Sleep(120);

}

}

// Hàm hiển thị menu chọn level

```
int disp_o_menu(int level){
```

```
    int pointer=0;
```

```
    for(;;){
```

```
        system("cls");
```

// Đặt màu cho console

```
        SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),2);
```

```
        cout<<"Select Level"<<endl;
```

// Đặt lại màu cho console

```
        SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),15);
```

```
        for(int i=0;i<level;i++)
```

```
        {
```

```
            if(i==pointer)
```

// Đặt màu cho console

```
            SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),11);
```

```
            cout<<"level "<<i+1<<endl;
```

// Đặt lại màu cho console

```
            SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),15);
```

```
        }
```

```
        for(int i=level;i<5;i++)
```

```
        {
```

// Đặt màu cho console

```
SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),8);  
cout<<"level "<<i+1<<endl;  
}
```

```
// Đặt lại màu cho console
```

```
SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),15);
```

```
// Xử lý sự kiện từ bàn phím
```

```
while(true)  
{  
    if(GetAsyncKeyState(VK_UP)!=0)  
    {  
        pointer-=1;  
        if(pointer== -1)  
            pointer=level-1;  
        break;  
    }  
}
```

```
// Xử lý sự kiện từ bàn phím
```

```
else if(GetAsyncKeyState(VK_DOWN)!=0)  
{  
    pointer+=1;  
    if(pointer==level)  
        pointer=0;  
    break;  
}
```

```
else if(GetAsyncKeyState(VK_RETURN)!=0)
{
    return pointer+1;
}
Sleep(120);
}
```

// Hàm chơi game

```
int game_play(int level[][4],int correct,int l)
{
```

// Khởi tạo các biến

```
int win=correct,ptr1,ptr2;
```

// Vòng lặp chạy cho đến khi win bằng 16

```
for(;win!=16;){
```

```
    system("cls");
```

```
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),2);
```

```
    cout<<"Game level " <<l;
```

```
    cout<<endl<<endl;
```

```
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),15);
```

// Hiển thị mảng level

```
    for(int i=0;i<4;i++)
```

```
    {
```

```
        for(int j=0;j<4;j++){
```

```

        if(level[i][j]==99)
        {

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),11);

        cout<<setw(3)<<"__";
        ptr1=i;ptr2=j;

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),15);

        }
        else
            cout<<setw(3)<<level[i][j];
    }
    cout<<endl;
}

cout<<"correct positions="<<setw(2)<<win<<" Incorrect="
"<<setw(2)<<16-win;

// Xử lý sự kiện từ bàn phím
while(true){
    if(GetAsyncKeyState(VK_UP)!=0)
    {
        if(ptr1>0&&ptr1<=3)
        {
            level[ptr1][ptr2]=level[ptr1-1][ptr2];
            level[ptr1-1][ptr2]=99;

```

```
        ptr1--;  
    }  
    break;  
}  
else if(GetAsyncKeyState(VK_DOWN)!=0)  
{  
    if(ptr1 >= 0 && ptr1 < 3){  
        level[ptr1][ptr2] = level[ptr1 + 1][ptr2];  
        level[ptr1 + 1][ptr2] = 99;  
        ptr1++;  
    }  
    break;  
}  
else if(GetAsyncKeyState(VK_LEFT)!=0)  
{  
    if(ptr2 > 0 && ptr2 <= 3){  
        level[ptr1][ptr2] = level[ptr1][ptr2 - 1];  
        level[ptr1][ptr2 - 1] = 99;  
        ptr2--;  
    }  
    break;  
}  
else if(GetAsyncKeyState(VK_RIGHT)!=0){
```

```

        if(ptr2>=0&&ptr2<3){
            level[ptr1][ptr2]=level[ptr1][ptr2+1];
            level[ptr1][ptr2+1]=99;
            ptr2++;
        }
        break;
    }
    else if(GetAsyncKeyState(VK_ESCAPE)!=0){
        return 1000;
    }
}

win=0;
// Kiểm tra số vị trí đúng
for(int i=0;i<4;i++)
{
    for(int j=0;j<4;j++)
    {
        if(level[i][j]==board[i][j])
            win++;
    }
}

Beep(1000,50);
Sleep(200);

```

```
}  
  
// Hiển thị thông báo khi hoàn thành game  
  
system("cls");  
cout<<endl<<"P";Sleep(100);  
cout<<"u";Sleep(100);  
cout<<"z";Sleep(200);  
cout<<"z";Sleep(200);  
cout<<"l";Sleep(200);  
cout<<"e ";Sleep(200);  
cout<<"c";Sleep(200);  
cout<<"o";Sleep(200);  
cout<<"m";Sleep(200);  
cout<<"p";Sleep(200);  
cout<<"l";Sleep(200);  
cout<<"e";Sleep(200);  
cout<<"t";Sleep(200);  
cout<<"e";Sleep(200);  
cout<<"d";Sleep(200);  
return 0;  
  
}
```

---Bài Report Đến Đây Là Kết Thúc---