**Trần Hoàng Triển - MSSV:46.01.104.196**

**Cài đặt bài toán 8 Quân Hậu**

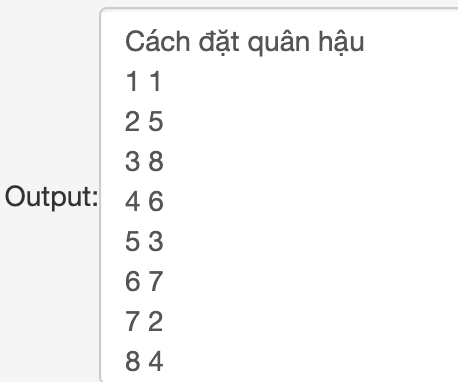
**Dữ liệu vào:**

**-** n là số quân hậu với bài này **n=8**

****

**Dữ liêu ra :**

Vị trí các quân hậu trong bàn cờ thoả mãn điều kiện sao cho không có 2 quân hậu nào ở cùng một hàng, cột hay đường chéo.



**Miêu tả cách xử lý dữ liệu như sau:**

1. Sau khi nhập n = 8. Biến `n` lưu trữ số lượng quân hậu cần đặt, mảng `mark` được sử dụng để đánh dấu các vị trí bị chiếm trên bàn cờ, và vector `res` được sử dụng để lưu trữ kết quả (các cột tương ứng với từng hàng).

2. Sau đó, chương trình gọi hàm `Try(1)` để bắt đầu quá trình đặt quân hậu trên hàng đầu tiên (hàng 1).

Hàm `Try(int row)` được định nghĩa để đặt quân hậu trên hàng `row`.

Nếu `row` đã vượt quá số lượng hàng (`row == n + 1`), nghĩa là đã đặt được tất cả quân hậu, chương trình in ra kết quả và kết thúc bằng lệnh `exit(0)`.

3. Trong hàm `Try`, vòng lặp `for` sẽ thử đặt quân hậu từ cột 1 đến cột 8 trên hàng hiện tại (hàng 1), chương trình kiểm tra xem có thể đặt quân hậu tại hàng `row`, cột `col` hay không. Điều kiện kiểm tra được thực hiện bằng cách sử dụng mảng `**mark`** để kiểm tra các cột, đường chéo chính và đường chéo phụ có bị chiếm hay không.

4. Nếu điều kiện đặt quân hậu thoả mãn (không bị chiếm), chương trình tiến hành đánh dấu cột, đường chéo chính và đường chéo phụ đã bị chiếm bằng cách đặt `mark[0][col]`, `mark[1][mainDiagonal]` và `mark[2][subDiagonal]` thành `true`. Sau đó, chương trình lưu cột `col` vào vector `res` và gọi đệ quy `Try(row+1)` để đặt quân hậu trên hàng tiếp theo.

5. Sau khi đặt quân hậu, chương trình đánh dấu cột, đường chéo chính và đường chéo phụ tương ứng đã bị chiếm.

6. Tiếp theo, chương trình thực hiện đệ quy gọi hàm `Try(row+1)` để đặt quân hậu trên hàng tiếp theo (hàng 2).

7. Quá trình đệ quy tiếp tục cho đến khi đặt quân hậu trên hàng cuối cùng (hàng 8). Khi đó, một cách đặt quân hậu hoàn chỉnh đã được tìm thấy.

8. Khi đã tìm thấy một cách đặt quân hậu hoàn chỉnh, chương trình in ra cách đặt đó và kết thúc bằng lệnh `exit(0)`.

9. Nếu không tìm thấy cách đặt quân hậu nào, chương trình sẽ kết thúc và không in ra kết quả.

**Để thực thi chương trình trên, bạn có thể làm theo các bước sau:**

1. Mở trình biên dịch C++ (như Dev-C++, Visual Studio,).

2. Tạo một tệp tin mới và sao chép đoạn mã vào tệp tin đó.

3. Lưu tệp tin với phần mở rộng `.cpp` (ví dụ: `**8QuanHau.cpp**`).

4. Tiến hành chạy chương trình và nhập dữ liệu từ bàn phím ‘n’: số quân hậu .

5. Trình biên dịch sẽ thực thi chương trình và hiển thị kết quả.

**Source code:**

#include<bits/stdc++.h>

using namespace std;

const int MaxN = 1 + 1e5;

int n;

bool mark[3][MaxN];

vector<int> res;

void Try(int row){

if(row == n + 1){

for(int i = 0 ; i < n ; ++i) cout << i + 1 << " " << res[i] << endl;

exit(0);

// Khi chỉ cần in ra 1 đáp án nên ta sẽ thoát khỏi chương trình ngay tại đây.

// Nếu cần tất cả đáp án ta sẽ không dùng exit(0)

}

for(int col = 1 ; col <= n ; ++col){

// Do hiệu của chỉ số hàng và cột có thể âm nên ta cộng thêm n

int mainDiagonal = row - col + n, subDiagonal = row + col;

// mark[0][]: cột

// mark[1][]: đường chéo chính

// mark[2][]: đường chéo phụ

if(mark[0][col] || mark[1][mainDiagonal] || mark[2][subDiagonal]) continue;

mark[0][col] = 1;

mark[1][mainDiagonal] = 1;

mark[2][subDiagonal] = 1;

res.push\_back(col);

Try(row + 1);

res.pop\_back();

mark[0][col] = 0;

mark[1][mainDiagonal] = 0;

mark[2][subDiagonal] = 0;

}

}

int main(){

cout << "Cách đặt quân hậu "<< endl;

cin >> n;

Try(1);

return 0;

}