

Tên: Thái Quốc Bảo

MSSV: 3123410031

```
#include <iostream>
#include <cstdlib> // cho rand(), srand()
#include <ctime>
#include <vector>
#include <algorithm>

using namespace std;

// TH6-01.CPP
// Tạo file ngẫu nhiên taci.txt; có cấu trúc như sau:
// -Dòng đầu ghi số n (với  $3 \leq n \leq 99$ ).
// -Trong n dòng tiếp theo; mỗi dòng ghi n số; các số cách nhau ít nhất một khoảng trắng. Yêu
// cầu giá trị các số từ 0 đến  $n^2 - 1$  và mỗi giá trị chỉ xuất hiện một lần trên bản TACI.
```

Tabnine | Edit | Test | Explain | Document

```
int main()
{
    srand(time(NULL));
    int n = 3 + rand() % 97; //  $3 \leq n \leq 99$ 
    cout << n << endl;
    vector<int> a;
    for (int i = 0; i < n * n; ++i)
    {
        a.push_back(i);
    }
    random_shuffle(a.begin(), a.end());
    freopen("taci.txt", "w", stdout);
    cout << n << "\n";
    for (int i = 0; i < n * n; ++i)
    {
        cout << a[i] << " ";
        if ((i + 1) % n == 0)
            cout << endl;
    }
    return 0;
}
```

```
// TH6-02.CPP
// Với 2 trạng thái bắt đầu và kết thúc được cho ngẫu nhiên.
// -Hãy tính giá trị độ đo ước lượng H1 của trạng thái ban đầu đối với trạng thái kết thúc.
// -Hãy tính giá trị độ đo ước lượng H2 của trạng thái ban đầu đối với trạng thái kết thúc.
// Ghi chú về H1 và H2:
// H1:
// Với độ ước lượng Trong đó = 0 nếu ai=bi và =1 nếu ai khác bi
// H2:
// Với hàm heuristic ; trong đó là số bước dịch chuyển (theo chiều ngang và
// chiều dọc) để đẩy ô ai về đúng vị trí ô bi.
```

```
#include <iostream>
#include <vector>
using namespace std;
```

```
#define pb push_back
```

```
class status
```

```
{
```

```
private:
```

```
    vector<vector<int>>> a;
```

```
    int n;
```

```
public:
```

```
    Tabnine | Edit | Test | Explain | Document
```

```
    status()
```

```
{
```

```
}
```

```
    Tabnine | Edit | Test | Explain | Document
```

```
    status(const int &n, const vector<vector<int>>> &a)
```

```
{
```

```
        this->n = n;
```

```
        this->a = a;
```

```
}
```

```
    Tabnine | Edit | Test | Explain | Document
```

```
    int getN() const
```

```
{
```

```
        return n;
```

```
}
```

```
vector<vector<int>> getA() const
{
    return a;
}
```

Tabnine | Edit | Test | Explain | Document

```
void setN(const int &n)
{
    this->n = n;
}
```

Tabnine | Edit | Test | Explain | Document

```
void setA(const vector<vector<int>> &a)
{
    this->a = a;
}
```

Tabnine | Edit | Test | Explain | Document

```
void input(const int &n)
{
    setN(n);
    vector<vector<int>> a;
    for (int i = 0; i < n; ++i)
    {
        vector<int> row;
        for (int j = 0; j < n; ++j)
        {
            int tmp;
            cin >> tmp;
            row.pb(tmp);
        }
        a.pb(row);
    }
    setA(a);
}
```

```
void print() const
{
    for (auto row : a)
    {
        for (int i : row)
        {
            cout << i << " ";
        }
        cout << "\n";
    }
}
```

// H1

Tabnine | Edit | Test | Explain | Document

```
int H1(const status &goal)
```

```
{
    int d = 0;
    for (int i = 0; i < goal.getN(); ++i)
        for (int j = 0; j < goal.getN(); ++j)
            if (this->a[i][j] != 0 && this->a[i][j] != goal.a[i][j])
                d++;
    return d;
}
```

```

void timViTri(const status &goal, const int &i, const int &j, int &x, int &y)
{
    for (int k = 0; k < goal.getN(); ++k)
    {
        for (int l = 0; l < goal.getN(); ++l)
        {
            if (this->a[i][j] == goal.a[k][l])
            {
                x = k;
                y = l;
                return;
            }
        }
    }
}

```

Tabnine | Edit | Test | Explain | Document

```

int H2(const status &goal)
{
    int x, y, d = 0;
    for (int i = 0; i < this->getN(); ++i)
    {
        for (int j = 0; j < this->getN(); ++j)
        {
            if (this->a[i][j] != 0)
            {
                timViTri(goal, i, j, x, y);
                d += abs(x - i) + abs(y - j);
            }
        }
    }
    return d;
}
};

```

```
int main()
{
    freopen("taci.txt", "r", stdin);
    int n;
    cin >> n;

    status start;
    start.input(n);

    status goal;
    goal.input(n);
    // H1
    cout << start.H1(goal) << endl;
    // H2
    cout << start.H2(goal) << endl;
    return 0;
}
```