**Phân tích từng phần của chương trình**

**1. Nhập dữ liệu**

int t; cin >> t;

while(t--){

cin >> n >> k;

int mt[n + 1][n + 1];

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

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

cin >> mt[i][j];

}

}

* t là số lượng bộ test.
* n là kích thước ma trận vuông n x n, k là giá trị tổng cần tìm.
* mt[i][j] là ma trận chứa giá trị tại hàng i, cột j.

**2. Sinh các hoán vị của dãy {1, 2, ..., n}**

vector<vector<int> > v;

// cấu hình đầu tiên

for(int i = 1; i <= n; i++) a[i] = i;

* v là danh sách lưu các hoán vị thỏa mãn điều kiện.
* a[i] là mảng khởi tạo hoán vị ban đầu {1, 2, ..., n}.

**3. Kiểm tra từng hoán vị**

do {

int sum = 0;

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

sum += mt[i][a[i]];

}

* next\_permutation(a + 1, a + n + 1) giúp sinh tất cả các hoán vị của {1, 2, ..., n}.
* Với mỗi hoán vị a, ta tính tổng sum:
  + sum += mt[i][a[i]] nghĩa là lấy phần tử từ **hàng i và cột a[i]** trong ma trận.

**4. Lưu các hoán vị thỏa mãn**

if(sum == k){

vector<int> tmp(a + 1, a + n + 1);

v.push\_back(tmp);

}

} while (next\_permutation(a + 1, a + n + 1));

* Nếu tổng sum == k, ta lưu hoán vị vào danh sách v.

**5. Xuất kết quả**

cout << v.size() << endl;

for(auto it : v){

for(int x : it){

cout << x << " ";

}

cout << endl;

}

* In số lượng hoán vị tìm được.
* In từng hoán vị thỏa mãn điều kiện.

Hãy cùng phân tích đầu vào và tìm kết quả của chương trình với test case:

**Input**

Sao chépChỉnh sửa

1

3 10

2 4 3

1 3 6

4 2 4

**Bước 1: Xây dựng ma trận**

Ma trận mt được nhập vào:

1 2 3 (Cột)

1 [2 4 3]

2 [1 3 6]

3 [4 2 4]

Chúng ta cần tìm các hoán vị của {1, 2, 3} sao cho tổng:

sum=mt[1][a1]+mt[2][a2]+mt[3][a3]=10

**Bước 2: Sinh tất cả các hoán vị của {1, 2, 3}**

Các hoán vị của {1, 2, 3} là:

1. {1, 2, 3}
2. {1, 3, 2}
3. {2, 1, 3}
4. {2, 3, 1}
5. {3, 1, 2}
6. {3, 2, 1}

**Bước 3: Tính tổng cho từng hoán vị**

|  |  |  |
| --- | --- | --- |
| **Hoán vị {a1, a2, a3}** | **sum = mt[1][a1] + mt[2][a2] + mt[3][a3]** | **Thỏa mãn sum == 10?** |
| {1, 2, 3} | mt[1][1] + mt[2][2] + mt[3][3] = 2 + 3 + 4 = 9 | ❌ |
| {1, 3, 2} | mt[1][1] + mt[2][3] + mt[3][2] = 2 + 6 + 2 = 10 | ✅ **Lưu lại** |
| {2, 1, 3} | mt[1][2] + mt[2][1] + mt[3][3] = 4 + 1 + 4 = 9 | ❌ |
| {2, 3, 1} | mt[1][2] + mt[2][3] + mt[3][1] = 4 + 6 + 4 = 14 | ❌ |
| {3, 1, 2} | mt[1][3] + mt[2][1] + mt[3][2] = 3 + 1 + 2 = 6 | ❌ |
| {3, 2, 1} | mt[1][3] + mt[2][2] + mt[3][1] = 3 + 3 + 4 = 10 | ✅ **Lưu lại** |

**Bước 4: Xuất kết quả**

Có **2 hoán vị** thỏa mãn điều kiện.

**Output**

2

1 3 2

3 2 1

**Giải thích kết quả**

* **Hoán vị {1, 3, 2}**: Chọn mt[1][1] = 2, mt[2][3] = 6, mt[3][2] = 2, tổng 2 + 6 + 2 = 10.
* **Hoán vị {3, 2, 1}**: Chọn mt[1][3] = 3, mt[2][2] = 3, mt[3][1] = 4, tổng 3 + 3 + 4 = 10.

**Tóm tắt quy trình**

1. **Đọc input**: Nhập n, k và ma trận mt.
2. **Sinh tất cả hoán vị của {1, 2, ..., n}**.
3. **Tính tổng** theo hoán vị và kiểm tra sum == k.
4. **Lưu các hoán vị hợp lệ và xuất kết quả**.

**Tóm tắt**

1. **Sinh tất cả các hoán vị** của {1, 2, ..., n}.
2. **Tính tổng** của các phần tử được chọn trong ma trận.
3. **Kiểm tra điều kiện** sum == k, nếu đúng thì lưu hoán vị.
4. **Xuất kết quả**.