# Bài Tập 4 Xử Lý Ma Trận

Cho ma trận hai chiều m dòng n cột, các phần tử là các số nguyên dương.  
1. Tìm số nguyên tố lớn nhất trong ma trận.  
2. Tìm những dòng của ma trận có chứa giá trị nguyên tố.  
3. Tìm những dòng của ma trận chỉ chứa các số nguyên tố.

## 1. Tìm số nguyên tố lớn nhất trong ma trận

Ý tưởng: Duyệt qua từng phần tử của ma trận, kiểm tra số nguyên tố và tìm số lớn nhất.

Code C++:

bool isPrime(int n) {  
 if (n < 2) return false;  
 for (int i = 2; i <= sqrt(n); i++) {  
 if (n % i == 0) return false;  
 }  
 return true;  
}  
  
int findMaxPrime(int arr[][100], int m, int n) {  
 int maxPrime = -1;  
 for (int i = 0; i < m; i++) {  
 for (int j = 0; j < n; j++) {  
 if (isPrime(arr[i][j]) && arr[i][j] > maxPrime) {  
 maxPrime = arr[i][j];  
 }  
 }  
 }  
 return maxPrime;  
}

## 2. Tìm những dòng có chứa số nguyên tố

Ý tưởng: Kiểm tra từng phần tử trong mỗi dòng, nếu có ít nhất một số nguyên tố thì in ra dòng đó.

Code C++:

void findRowsWithPrime(int arr[][100], int m, int n) {  
 for (int i = 0; i < m; i++) {  
 bool hasPrime = false;  
 for (int j = 0; j < n; j++) {  
 if (isPrime(arr[i][j])) {  
 hasPrime = true;  
 break;  
 }  
 }  
 if (hasPrime) {  
 cout << "Dòng " << i + 1 << " chứa số nguyên tố\n";  
 }  
 }  
}

## 3. Tìm những dòng chỉ chứa các số nguyên tố

Ý tưởng: Kiểm tra toàn bộ các phần tử trong một dòng, nếu tất cả đều là số nguyên tố thì in ra dòng đó.

Code C++:

void findRowsAllPrime(int arr[][100], int m, int n) {  
 for (int i = 0; i < m; i++) {  
 bool allPrime = true;  
 for (int j = 0; j < n; j++) {  
 if (!isPrime(arr[i][j])) {  
 allPrime = false;  
 break;  
 }  
 }  
 if (allPrime) {  
 cout << "Dòng " << i + 1 << " chỉ chứa toàn số nguyên tố\n";  
 }  
 }  
}