TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP. HCM

**KHOA CÔNG NGHỆ THÔNG TIN**



BÁO CÁO TIỂU LUẬN

**Đề Tài: *Số 4***

**Nhóm : 1**

**Sinh Viên Thực Hiện:** Nguyễn Minh Thư

Nguyễn Công Dũng

Nguyên Huy Hoàng

Lê Thị Khánh Vân

Phan Văn Vũ

TP. Hồ Chí Minh, năm 2022

#### MỤC LỤC

[MỤC LỤC i](#_Toc94203826)

[DANH MỤC HÌNH ẢNH iii](#_Toc94203827)

[Phần 1: Giới thiệu về bài báo cáo: 1](#_Toc94203828)

[1.1. Nội dung tìm hiểu: 1](#_Toc94203829)

[1.1.1. Mảng 2 chiều: 1](#_Toc94203830)

[1.1.2. Mảng câu trúc: 1](#_Toc94203831)

[1.2. Bảng phân công công việc: 1](#_Toc94203832)

[Phần 2: Mảng 2 chiều 2](#_Toc94203833)

[2.1. Nhập/ xuất A: từ bàn phím, tạo ngẫu nhiên, từ file text. 2](#_Toc94203834)

[2.1.1. Tạo ma trận: 2](#_Toc94203835)

[2.1.2. Xuất ma trận: 3](#_Toc94203836)

[2.2. Tính giá trị trung bình cộng của các phần tử trên 4 biên, đường chéo chính và đường chéo phụ. 4](#_Toc94203837)

[2.3. Liệt kê danh sách phần tử của từng cột chứa toàn bộ nhữsố thịnh vượng. 6](#_Toc94203838)

[2.4. Hãy cho biết dòng có chứa giá trị trung bình nhân của các số nguyên tố lớn nhất. 8](#_Toc94203839)

[2.5. Hãy hoán vị hai cột c1 và c2 bất kỳ của ma trận. 10](#_Toc94203840)

[2.6. Từ A đã cho hãy tạo và xuất ra một ma trận B sao cho từng dòng phải được sắp xếp tất các giá trị chẵn tăng dần và các giá trị lẻ giảm dần. 11](#_Toc94203841)

[2.7. Từ A đã cho hãy tạo và xuất ra 2 ma trận C, D (có cùng kích thước) sao cho: C chứa toàn số dương và D chứa toàn số âm (các vị trí trống còn lại trên C, D để số 0). 13](#_Toc94203842)

[2.8. Tạo một ma trận E có kích thước là x dòng × y cột (với 0 < x < m, 0 < y < n). Hãy đếm số lần xuất hiện của E trong A. 14](#_Toc94203843)

[2.9. Từ A đã cho hãy tạo và xuất ra một ma trận F được dịch phải xoay vòng các cột theo trục đứng với chiều từ trái sang phải k lần (k > 0 được nhập từ bàn phím). 16](#_Toc94203844)

[2.10. Từ A đã cho hãy tạo và xuất ra một ma trận I được xoắn ốc giảm dần ngược chiều kim đồng hồ. 17](#_Toc94203845)

[Phần 3: Cấu trúc 20](#_Toc94203846)

[3.1. Tính tổng tiền của một đơn thuốc. 20](#_Toc94203847)

[3.2. Tính tổng tiền mà nhà thuốc đã thu được khi bán xong các đơn thuốc. 20](#_Toc94203848)

[3.3. Sắp xếp đơn thuốc tăng dần theo tổng tiền. 21](#_Toc94203849)

[3.4. Hãy liệt kê các đơn thuốc đã bán cho khách hàng tên là X 21](#_Toc94203850)

[3.5. Cập nhật lại thông tin tên bác sĩ trong don thuoc cua benhnhanX 22](#_Toc94203851)

[3.6. Thêm một loại thuốc vào đơn thuốc có mã X trong danh sách. 23](#_Toc94203852)

[3.7. Xóa các đơn thuốc mà bệnh nhân X đã mua. 24](#_Toc94203853)

[3.8. Cho biết loại thuốc nào được nhà thuốc bán ra nhiều nhất cùng với số lượng của nó. 25](#_Toc94203854)

[Ý tưởng thuật toán: 25](#_Toc94203855)

[3.9. Liệt kê các đơn thuốc của bác sĩ Y đã kê đơn. 27](#_Toc94203856)

[3.10. Tính hoa hồng mà tiệm thuốc nhận được khi bán các đơn thuốc. Biết rằng tiền hoa hồng của các hóa đơn được xác định dựa vào tổng tiền như sau: Nếu tổng tiền nhỏ hơn 1000 thì hoa hồng là 10%, nếu 1000<tổng tiền<=10000 thì hoa hồng là 8%, còn lại hoa hồng là 5%.. 27](#_Toc94203857)

[3.11. Cho biết loại thuốc nào được nhà thuốc bán ra nhiều nhất cùng với số lượng của nó. 29](#_Toc94203858)

[3.12. Thống kê theo bệnh nhân số tiền và số đơn thuôc họ đã mua tại nhà thuốc theo mẫu. 30](#_Toc94203859)

[Phần 4: Tổng kết - Đánh giá công việc nhóm: 32](#_Toc94203860)

[4.1. Công việc cá nhân: 32](#_Toc94203861)

[4.2. Đánh giá chung: 32](#_Toc94203862)

#### DANH MỤC HÌNH ẢNH

[Hình 1: Tạo ma trận bằng cách nhập từ bàn phím 2](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184338)

[Hình 2: Tạo ma trận ngẫu nhiên 3](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184339)

[Hình 3: Tạo ma trận với các giá trị đọc vào từ file 3](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184340)

[Hình 4: Xuất ma trận 4](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184341)

[Hình 5: Tính trung bình cộng của các phần tử trên 4 biên 5](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184342)

[Hình 6: Tính trung bình cộng của các phần tử trên đường chéo chính. 5](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184343)

[Hình 7: Tính trung bình cộng của các phần tử trên đường chéo phụ. 6](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184344)

[Hình 8: Kiểm tra số thịnh vượng 7](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184345)

[Hình 9: 7](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184346)

[Hình 10: In các cột có toàn số thịnh vượng 8](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184347)

[Hình 11: Kiểm tra số nguyên tố 8](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184348)

[Hình 12: Tính trung bình nhân của các giá trị là số nguyên tố trên dòng k 9](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184349)

[Hình 13: Tìm giá trị lớn nhất trung bình nhân các số nguyên tố của dòng 9](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184350)

[Hình 14: Xuất dòng có giá trị trung bình nhân các số nguyên tố lớn nhất 10](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184351)

[Hình 15: Hoán vị hai cột c1 và c2 bất kì của ma trận 11](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184352)

[Hình 16: Tạo ma trận A tương tự ma trận B 11](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184353)

[Hình 17: Hàm sắp xếp các phần tử chẵn trong dòng tăng dần 12](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184354)

[Hình 18: Hàm xắp xếp các phần tử lẻ trong dòng giảm dần 12](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184355)

[Hình 19: Sắp xếp ma trận với các phần tử chẵn tang dần và phần tử lẻ giảm dần 13](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184356)

[Hình 20: Tạo ma trận C chứa toàn số dương các giá trị khác bằng không 13](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184357)

[Hình 21: Tạo ma trận D chứa toàn số âm các giá trị khác bằng không 14](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184358)

[Hình 22: Tạo ma trận E (0 < x < row, 0 < y < col) 14](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184359)

[Hình 23: Đếm số lần ma trận E xuất hiện trong ma trận A 15](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184360)

[Hình 24: Đếm số lần ma trận E xuất hiện trong ma trận A 16](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184361)

[Hình 25: Ma Trận F được dịch phải xoay vòng các cột theo trục đứng với chiều từ trái sang phải k lần 17](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184362)

[Hình 26: Xắp xếp mảng 1 chiều 17](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184363)

[Hình 27: Tạo ma trận xoắn ốc 18](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184364)

[Hình 28: Tạo ma trận xoắn ốc 18](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184365)

[Hình 29: Tạo ma trận xoắn ốc 19](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184366)

[Hình 30: Tính tổng tiền một đơn thu 20](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184367)

[Hình 31: Tính tổng tiền mà nhà thuốc đã thu được khi bán xong các đơn thuốc 20](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184368)

[Hình 32: Sắp xếp đơn thuốc tăng dần theo tổng tiền. 21](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184369)

[Hình 33: Hãy liệt kê các đơn thuốc đã bán cho khách hàng tên là X 22](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184370)

[Hình 34: Cập nhật lại thông tin tên của bác sĩ trong đơn thuốc của bệnh nhân có tên là X 23](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184371)

[Hình 35: Thêm một loại thuốc có mã thuốc là X trong danh sách 24](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184372)

[Hình 36: Xóa một đơn thuốc mà bênh nhân X đã mua 25](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184373)

[Hình 37: Cho biết loại thuốc nào được nhà thuốc bán ra nhiều nhất cùng với số lượng của nó 26](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184374)

[Hình 38: Liệt kê các đơn thuốc mà bác sĩ Y đã kê đơn 27](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184375)

[Hình 39: Tính tiền hoa hồng của tiệm thuốc 28](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184376)

[Hình 40: Cho biết loại thuốc nào được bán ra nhiều nhất và số lượng của nó 30](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184377)

[Hình 41: Thống kê bệnh nhân và đơn thuốc của bệnh nhân đã mua 31](file:///C:\Users\KHANHVAN-PC\Downloads\12DHTH15-Nhóm-1-Đề-tài-số-4-1.docx#_Toc94184378)

# Giới thiệu về bài báo cáo:

## Nội dung tìm hiểu:

### Mảng 2 chiều:

Mảng 2 chiều là kiểu mảng chứa các mảng khác bên trong nó. Phần tử của mảng 2 chiều không được lưu giữ trực tiếp trong mảng 2 chiều, mà được lưu giữ thông qua các mảng 1 chiều bên trong mảng 2 chiều đó. Do cấu tạo mảng như vậy nên chúng ta mới gọi các mảng trong mảng như thế này là mảng 2 chiều.

### Mảng câu trúc:

Kiểu cấu trúc trong C++, hay còn gọi là kiểu struct trong C++ là một tập hợp các thuộc tính liên quan tới cùng một đối tượng. Ví dụ điển hình của kiểu cấu trúc là tập hợp các thuộc tính liên quan tới một người như tên, tuổi và giới tính.

## Bảng phân công công việc:

|  |  |  |
| --- | --- | --- |
| Thành viên | Mảng 2 chiều | Mảng cấu trúc |
| Nguyễn Minh Thư (Nhóm trưởng ) | Bài 9, 10 | Bài 1, 4, 15 |
| Nguyễn Huy Hoàng | Bài 1, 2 | Bài 2, 3, 5 |
| Nguyễn Công Dũng | Bài 3, 4 | Bài 6, 7, 8 |
| Phan Văn Vũ | Bài 5, 6 | Bài 9, 10, 11 |
| Lê Thị Khánh Vân | Bài 7, 8 | Bài 12, 13, 14 |

# Mảng 2 chiều

## Nhập/ xuất A: từ bàn phím, tạo ngẫu nhiên, từ file text.

### Tạo ma trận:

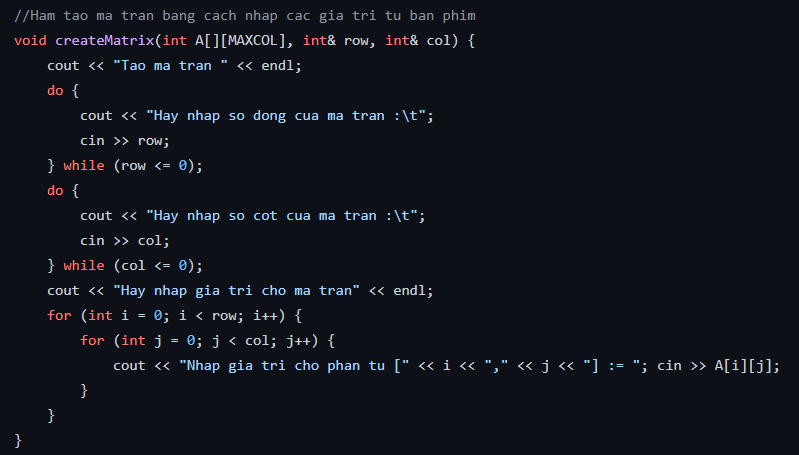
* Các phần tử nhập từ bàn phím:

Ý tưởng thuật toán:

+ Tạo 2 biến tương đương với dòng và cột của ma trận và gán giá trị cho 2 biến đó (Điều kiện > 0) và bắt nhập đến khi đúng thì dừng bằng while.

+ Sau khi có giá trị dòng và cột, lồng hàm vòng lặp for với 2 biến i ở ngoài và j ở trong.

+ Nhập giá trị từ bán phím khi hàm for thực hiện vòng lặp tương ứng với từng phần tử trong ma trận.

Code :

Hình : Tạo ma trận nhập vào từ bàn phím

Hình : Tạo ma trận bằng cách nhập từ bàn phím

* Ma trận ngẫu nhiên:

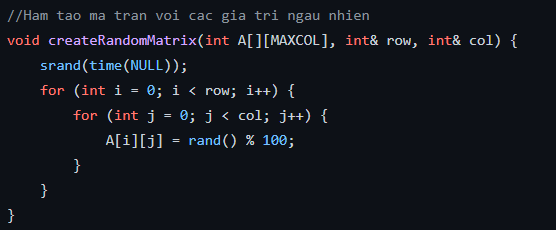
Ý tưởng thuật toán:

+ Tạo 2 biến tương đương với dòng và cột của ma trận và gán giá trị cho 2 biến đó (Điều kiện > 0) và bắt nhập đến khi đúng thì ngưng bằng while.

+ Sau khi có giá trị dòng và cột, lồng hàm vòng lặp for với 2 biến i ở ngoài và j ở trong.

+ Khi hàm for thực hiện vòng lặp tương ứng với từng phần tử trong ma trận cho từng phần tử bằng hàm ramdom để tạo ma trận ngẫu nhiên.

Vd: arr[i][j] = rand() % 100 ( giá trị ngẫu nhiên từ 0 đến 100 )

 Code :

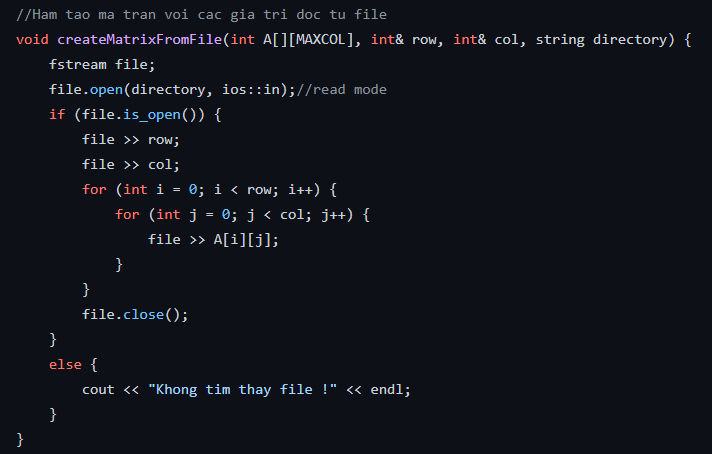
Hình : Tạo ma trận ngẫu nhiên

* Ma trận được nhập từ File:

Ý tưởng thuật toán:

+ Dùng thư viện fstream để thực hiện việc mở và đọc file liên kết.

+ Đọc các giá trị từ file gồm dòng, cột và giá trị các phần tử trong ma trận.

 Code :

Hình : Tạo ma trận với các giá trị đọc vào từ file

### Xuất ma trận:

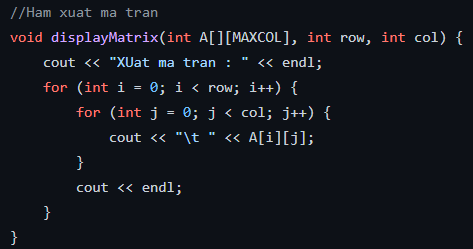
Ý tưởng thuật toán:

* Lồng hàm :

+ Chạy vòng lặp for với biến i tương đương với số hàng trong ma trận.

+ Chạy thêm 1 vòng lặp for bên trong với biến j tương đương với số phần tử trên mỗi hàng.

+ Xuất các phân tử đó ra. ( dùng " \t " hoặc " " để căn các phần tử )

 Code :

Hình : Xuất ma trận

## Tính giá trị trung bình cộng của các phần tử trên 4 biên, đường chéo chính và đường chéo phụ.

* Trung bình cộng của các phần tử trên 4 biên:

Ý tưởng thuật toán :

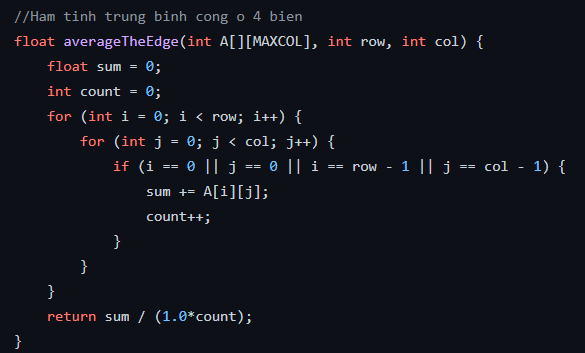
+ Đường biên của ma trận  :

i = 0 Hoặc i = n – 1 Hoặc j = 0 Hoặc j = n -1 ( arr[i][j] )

+ Tạo hàm tính tổng (sum), hàm đếm số lượng các phần tử trong ma trận (count).

+ Trong hàm lồng vòng lặp for: tính tổng các phần tử thõa điều kiện ở biên và đếm số lượng các phần tử đó.

+ Tính trung bình cộng ở ngoài hàm for ( sum / count )

 Code :

Hình : Tính trung bình cộng của các phần tử trên 4 biên

* Trung bình cộng của các phần tử trên đường chéo chính.

Ý tưởng thuật toán :

Điều kiện : Ma trận vuông ( row = col )

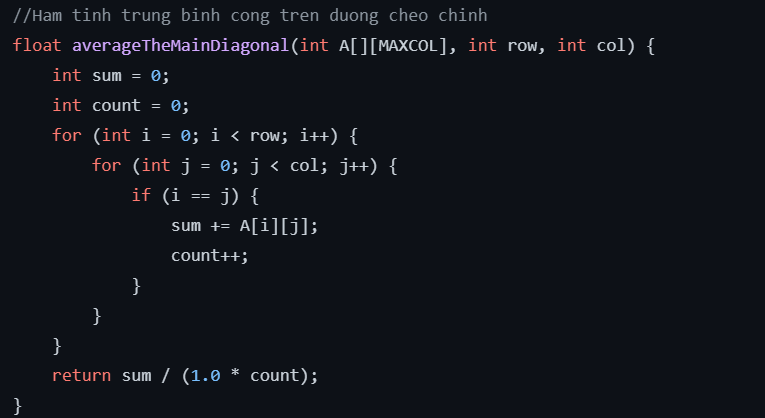
+ Đường chéo chính của ma trận vuông :

i = j ( arr[i][j] )

+ Tạo hàm tính tổng (sum), hàm đếm số lượng các phần tử trong ma trận (count).

+ Trong hàm lồng vòng lặp for: tính tổng các phần tử thõa điều kiện ở đường chéo chính và đếm số lượng các phần tử đó.

+ Tính trung bình cộng ở ngoài hàm for ( sum / count )

Code :

Hình : Tính trung bình cộng của các phần tử trên đường chéo chính.

* Trung bình cộng của các phần tử trên đường chéo phụ.

Ý tưởng thuật toán :

Điều kiện : Ma trận vuông ( row = col )

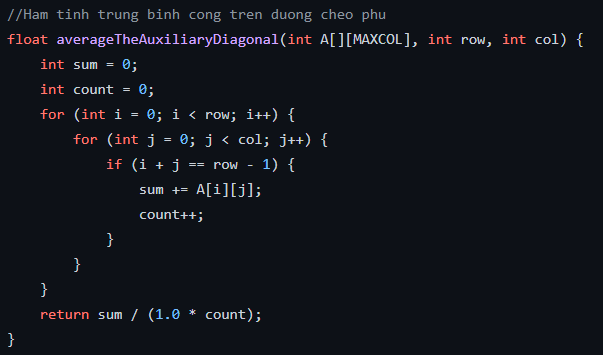
+ Đường chéo phụ của ma trận vuông :

i = n – 1 – j ( arr[i][j] )

+ Tạo hàm tính tổng (sum), hàm đếm số lượng các phần tử trong ma trận (count).

+ Trong hàm lồng vòng lặp for: tính tổng các phần tử thõa điều kiện ở đường chéo chính và đếm số lượng các phần tử đó.

+ Tính trung bình cộng ở ngoài hàm for ( sum / count )

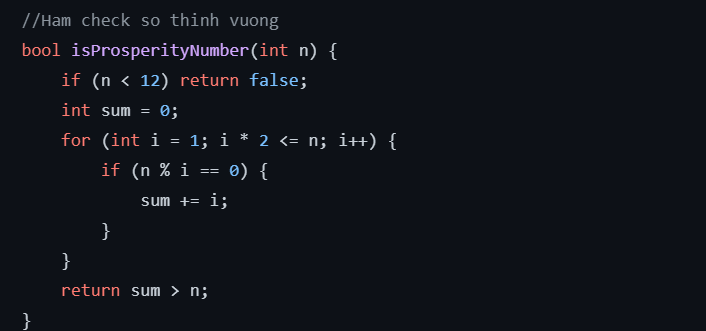
Code :

Hình 7: Tính trung bình cộng của các phần tử trên đường chéo phụ.

## Liệt kê danh sách phần tử của từng cột chứa toàn bộ nhữsố thịnh vượng.

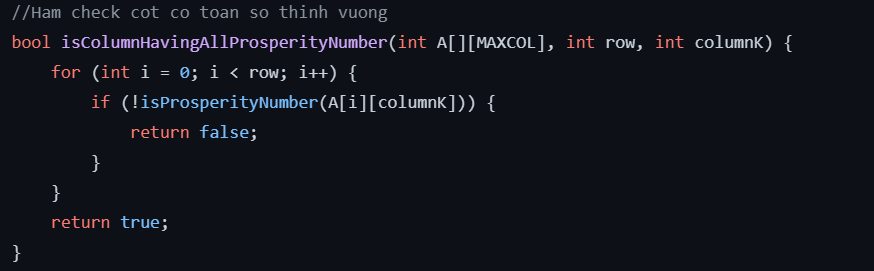
Ý tưởng thuật toán:

* Số thịnh vượng là số có tổng các ước không bao gồm số đó, lớn hơn nó.
* Hàm kiểm tra số thịnh vượng : vòng lặp for để tìm các ước của phẩn tử đó và tính tổng của các ước đó.

Code :

Hình : Kiểm tra số thịnh vượng

* Hàm kiểm tra cột có toàn số thịnh vượng: Cho chạy vòng lặp for từng phần tử trên cột với điều kiện dùng hàm kiểm tra số thịnh vượng ở trên. Nếu có 1 phần tử trong cột không phải số thịnh vượng lập tức qua cột mới trả về false Còn cột toàn là số thịnh vượng sẽ được trả về true.

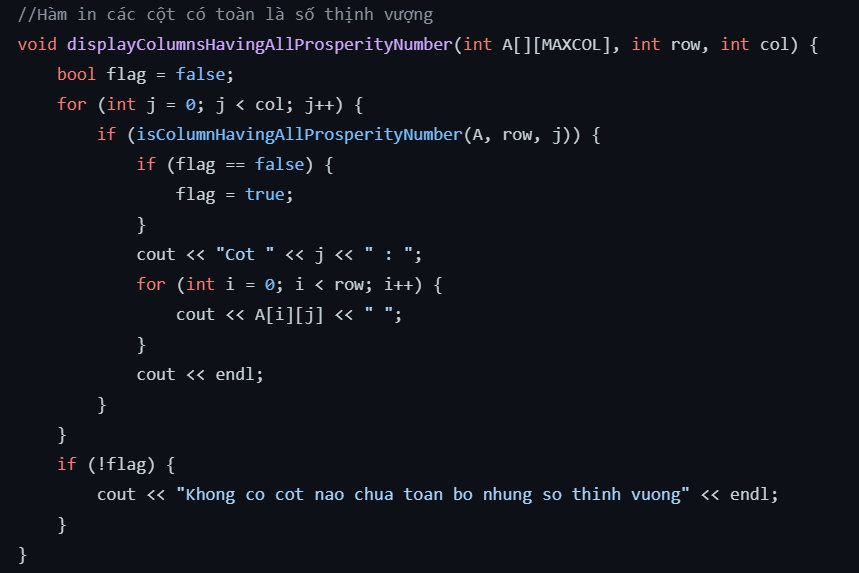
Code :

Hình :

* Hàm in các cột có toàn là số thịnh vượng: Duyệt vòng lặp for qua tất cả các cột, trong vòng lặp for kiểm tra điều kiện hàm check cột có toàn số thịnh vượng(isColumnHavingAllProsperityNumber), nếu điều kiện đúng thì in ra cột cùng với các phần tử của cột. Nếu điều kiện sai thì thoát khỏi vòng lặp và in ra thông báo không có cột nào toàn là số thịnh vượng.

Code :

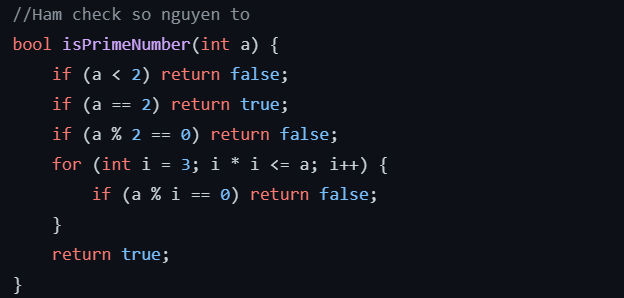
Hình : In các cột có toàn số thịnh vượng



## Hãy cho biết dòng có chứa giá trị trung bình nhân của các số nguyên tố lớn nhất.

Ý tưởng thuật toán :

* Tạo hàm kiểm tra số nguyên tố: Cho chạy vòng lặp for với 1 biến xuất phát từ 3 đến giá trị phần tử đó, phần tử nào không chia hết cho bất kỳ số nào của biến thì đó là số nguyên tố. (SNT lớn hơn hoặc bằng 2).

Code:

Hình : Kiểm tra số nguyên tố

* Tạo hàm để tính trung bình nhân của các số nguyên tố đã tìm trên từng dòng. Trong vòng lặp for xét nếu phần tử trên dòng đó là số nguyên tố sẽ được nhân lại với nhau (product) và đếm số lượng phần tử là số nguyên tố trên dòng đó (count ).

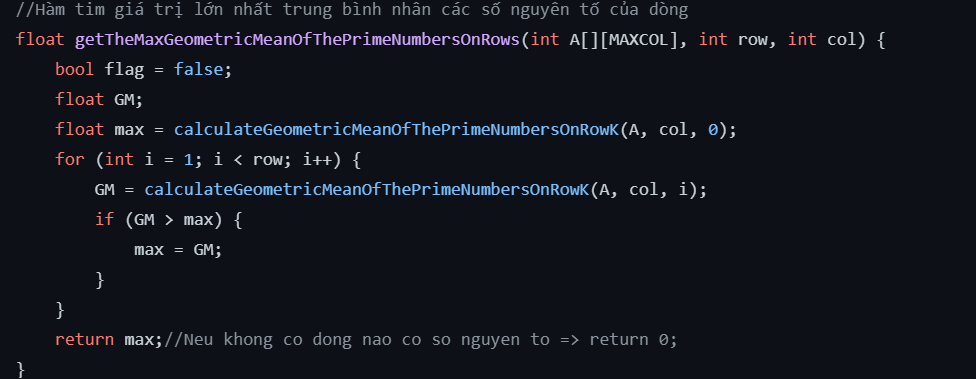
Sau đó trả về giá trị :

Code :

Hình : Tính trung bình nhân của các giá trị là số nguyên tố trên dòng k

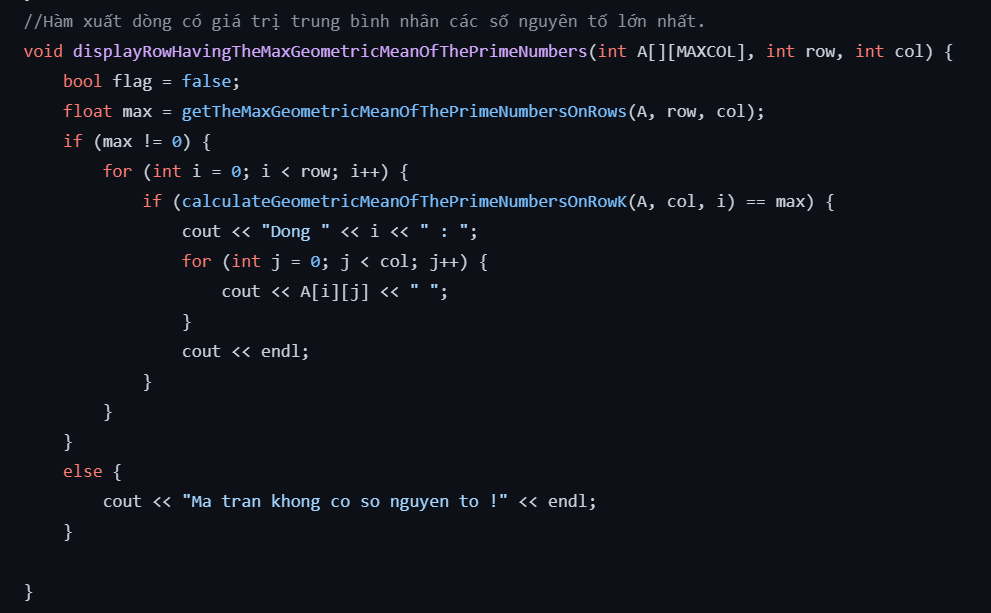
* Tạo hàm tìm dòng có giá trị trung bình nhân các số nguyên tố lớn nhất: Giả sử dòng đầu tiên lớn nhất (max) sau đó xét nếu dòng 2 lớn hơn dòng 1 thì sẽ cho max = giá trị của dòng 2 và ngược lại thì không. Giá trị trả về là max.

Code :



Hình : Tìm giá trị lớn nhất trung bình nhân các số nguyên tố của dòng

* Hàm xuất dòng có giá trị trung bình nhân các số nguyên tố lớn nhất: Khởi tạo một biến max gán cho bằng hàm tim giá trị lớn nhất trung bình nhân các số nguyên tố của dòng tiến hành kiểm tra nếu max khác không thì bắt đầu thực hiện vòng lặp for với điều kiện là trung bình nhân các số nguyên tố của dòng bằng với max thì xuất ra màn hình dòng đất ngược lại xuất ra thông báo : Ma trận không có số nguyên tố.

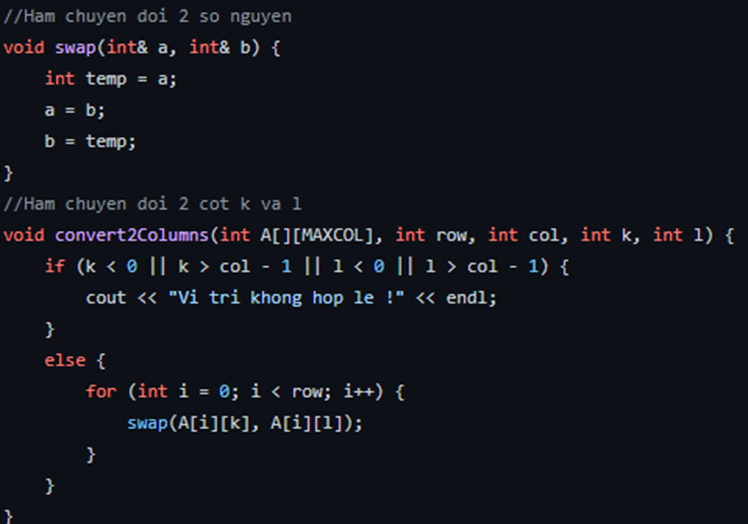
Code :

Hình : Xuất dòng có giá trị trung bình nhân các số nguyên tố lớn nhất

## Hãy hoán vị hai cột c1 và c2 bất kỳ của ma trận.

Ý tưởng thuật toán :

* Tạo 1 hàm hoán vị 2 số nguyên: tạo biến thứ 3 để hỗ trợ đổi chỗ 2 số nguyên cần hoán vị.
* Cho nhập từ bàn phím 2 cột cần hoán vị ( điều kiện phải nằm trong khoảng số cột và dòng ). Nếu thõa điều kiện, thực hiện việc chuyển đổi 2 cột bằng hàm hoán vị vừa tạo.

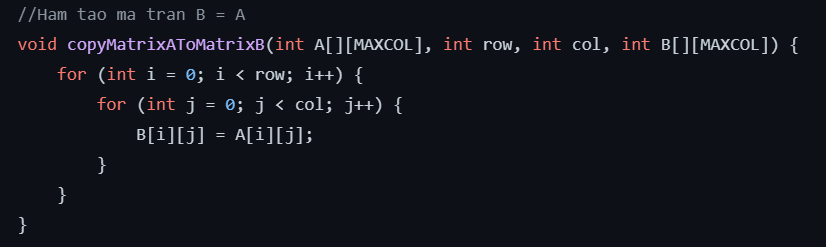
Code :

Hình : Hoán vị hai cột c1 và c2 bất kì của ma trận

## Từ A đã cho hãy tạo và xuất ra một ma trận B sao cho từng dòng phải được sắp xếp tất các giá trị chẵn tăng dần và các giá trị lẻ giảm dần.

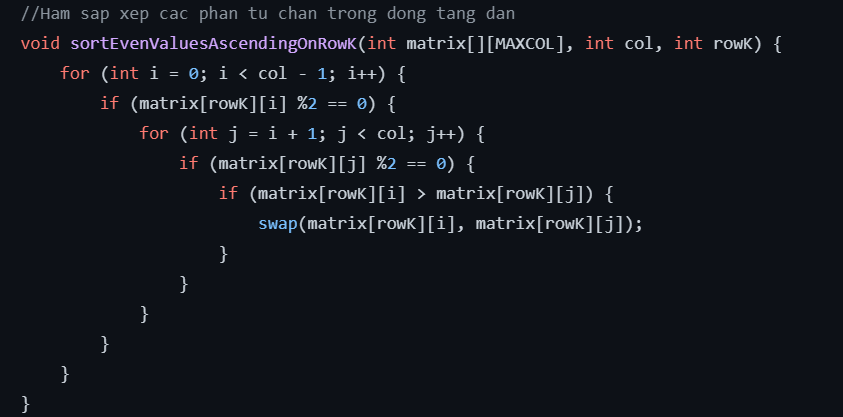
Ý tưởng thuật toán:

* Tạo 1 ma trận B tương tự ma trận A :

Code :

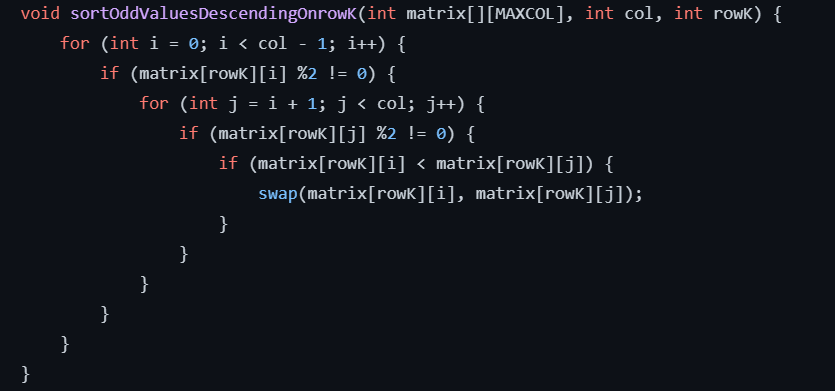
Hình 16: Tạo ma trận A tương tự ma trận B

* Trong ma trận B tạo 1 hàm sắp xếp các phần tử chẵn trong dòng tăng dần :
* Tìm phần chẵn đầu tiên trong dòng sau đó xét theo thứ tự các phần tử chẵn ở sau và chuyển đổi các phần tử theo giá trị tăng dần bằng hàm swap.

Code :

Hình : Hàm sắp xếp các phần tử chẵn trong dòng tăng dần

* Tạo một hàm tương tự hàm trên nhưng với số lẻ.

Code :

Hình : Hàm xắp xếp các phần tử lẻ trong dòng giảm dần

* Kết hợp hai hàm trên để xuất ra từng dòng với giá trị các phần tử chẵn tăng dần và các phần tử lẻ giảm dần.

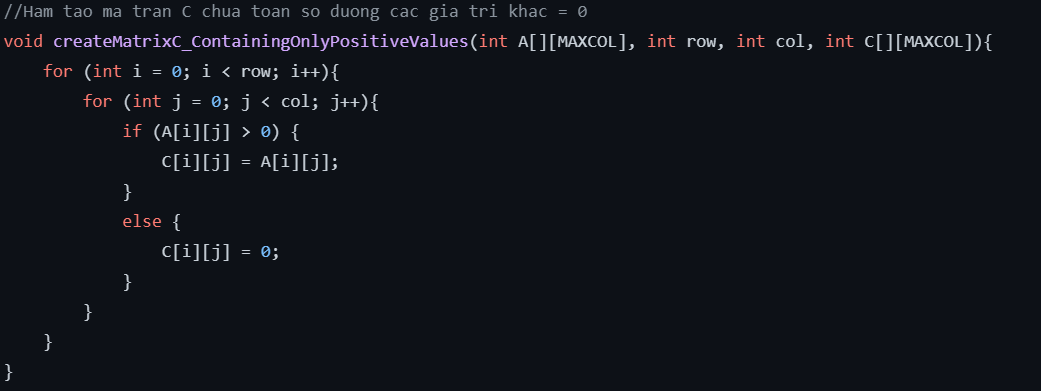
Code :

Hình : Sắp xếp ma trận với các phần tử chẵn tang dần và phần tử lẻ giảm dần

## Từ A đã cho hãy tạo và xuất ra 2 ma trận C, D (có cùng kích thước) sao cho: C chứa toàn số dương và D chứa toàn số âm (các vị trí trống còn lại trên C, D để số 0).

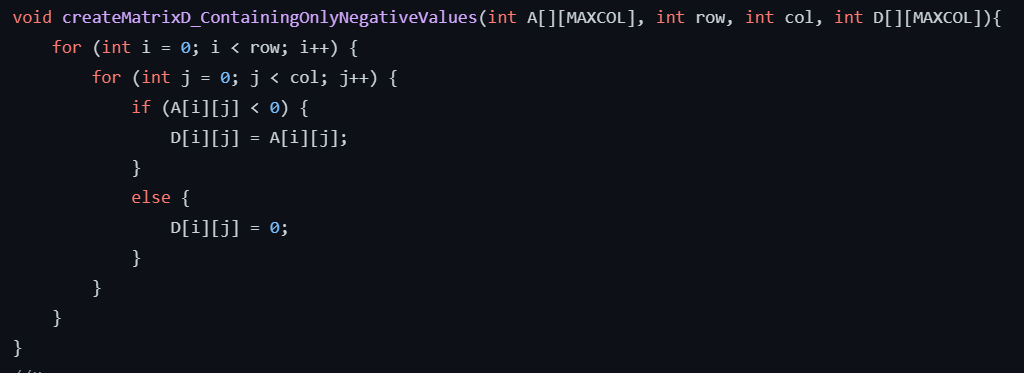
Ý tưởng thuật toán :

* Tạo ma trận C với các phần toàn số dương : Sử dụng số dòng và cột có sẵn ở ma trận A sử dụng vòng lặp for. Nếu giá trị phần tử của A lớn hơn không (A[i][j] > 0) thì phần tử ấy sẽ được giữ nguyên giá trị đưa vào ma trận C, ngược lại nếu giá trị phần tử của A nhỏ hơn không (A[i][j] < 0) thì tại ma trận C sẽ được ghi vào một con số không (C[i][j] = 0)

Code :

Hình : Tạo ma trận C chứa toàn số dương các giá trị khác bằng không

* Tạo ma trận D với các phần toàn số âm : Sử dụng số dòng và cột có sẵn ở ma trận A cho lồng vòng lặp for. Nếu giá trị phần tử của A nhỏ hơn không (A[i][j] < 0) sẽ được gán và giữ nguyên và đưa vào ma trận D, , ngược lại nếu giá trị phần tử của A nhỏ hơn không (A[i][j] > 0) thì tại ma trận C sẽ được ghi vào một con số không (C[i][j] = 0)

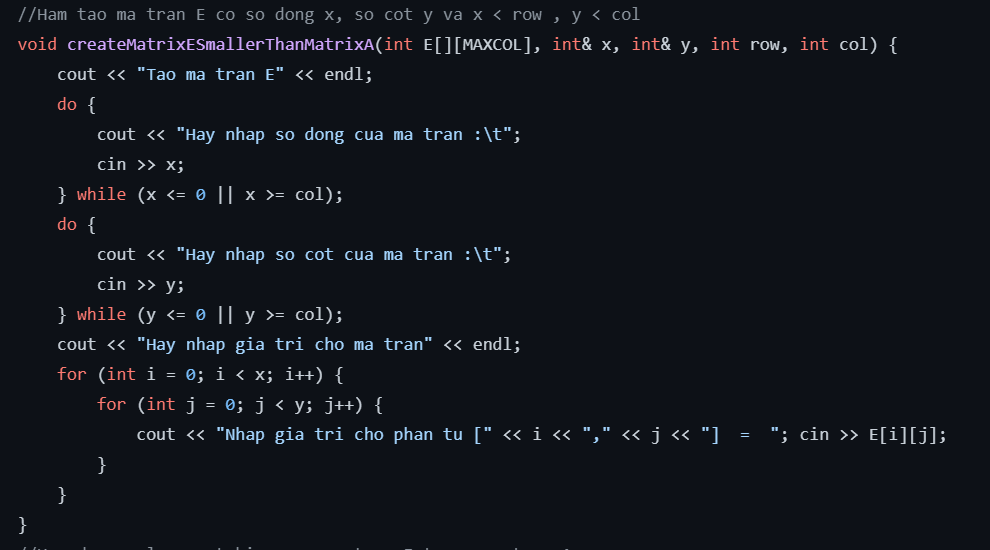
Code :

Hình : Tạo ma trận D chứa toàn số âm các giá trị khác bằng không

## Tạo một ma trận E có kích thước là x dòng × y cột (với 0 < x < m, 0 < y < n). Hãy đếm số lần xuất hiện của E trong A.

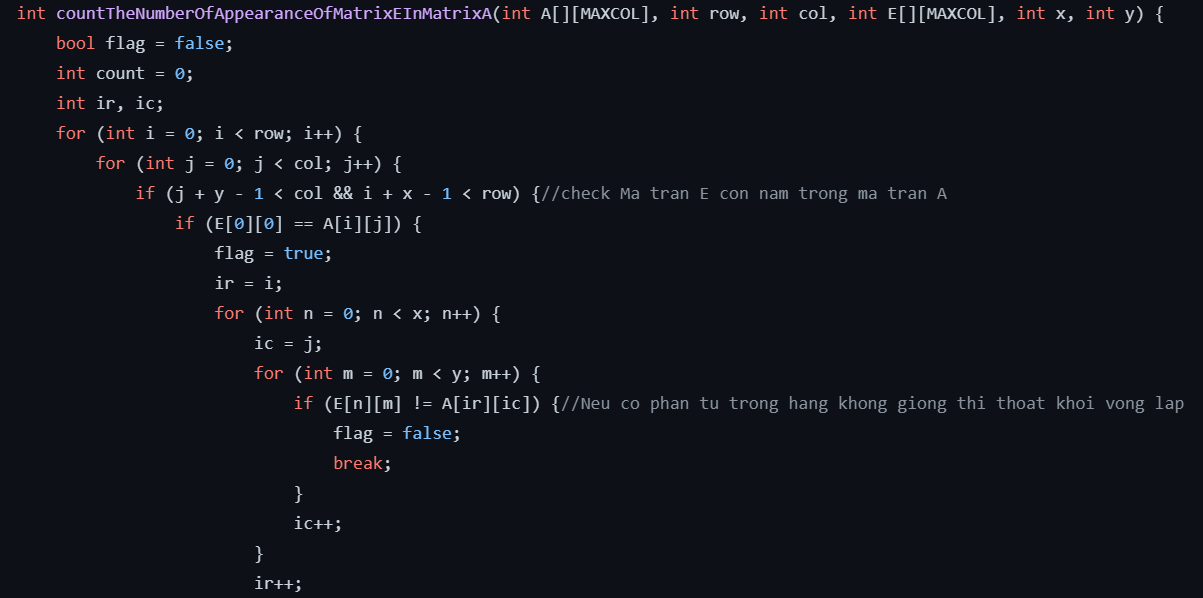
Ý tưởng thuật toán :

* Tạo ma trận E với số hàng ( x ) và cột ( y ) nhập từ bàn phím ( điều kiện số dòng và cột của ma trận E phải nhỏ hơn ma trận A ) và nhập giá trị từng phần tử trong ma trận E.

Code :

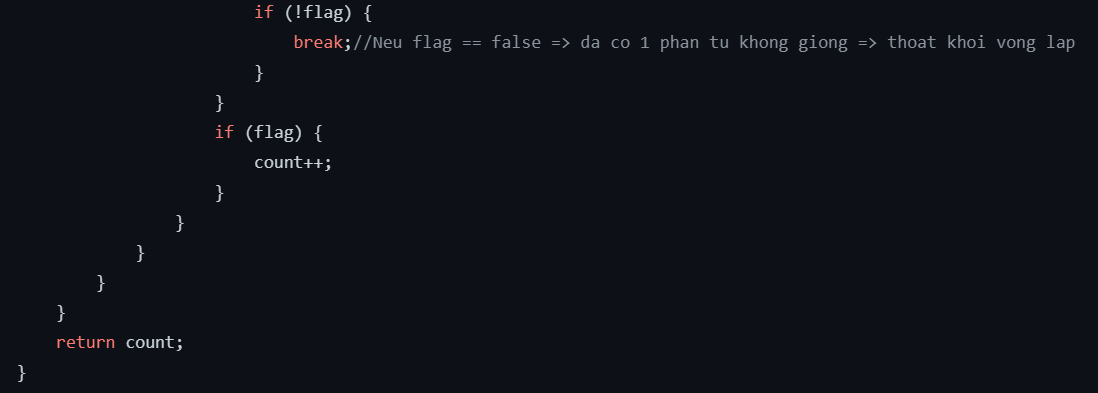
Hình 22: Tạo ma trận E (0 < x < row, 0 < y < col)

* Hàm đếm số lần xuất hiện của ma trận A trong ma trận A:
* Trong vòng lặp for ( i và j ) với điều kiện j + y – 1 < col ( của A ) và i + x – 1 < row( của A ) để kiểm tra những phần tử đầu tiên của ma trận A thõa mãn i và j trong điều kiện với phần tử đầu tiên của ma trận E. Nếu không thõa điều kiện i và j ở trên thì khi xét ma trận E sẽ nằm ngoài kích thước của ma trận A. Vd: Phần tử đầu tiên của ma trận E giống với phần tử cuối cùng của ma trận A => Không thể thực hiện vòng lặp để xét tiếp.
* Sau khi thõa điều kiện E[0][0] = A[i][j] như trên, ta xét xuất phát từ A[i][j] với kích thước ma trận E, tương đương vị trí từng phần tử của ma trận E với từng phần tử của ma trận A. Nếu có 1 phần tử khác nhau ở vị trí đó lập tức thoát vòng lặp và ngược lại nếu tất cả phần tử được xét giống nhau. Và thực hiện tiếp vòng lặp. Đặt 1 biến đếm ( count ), khi tìm được ma trận E trong ma trận A thì biến đếm sẽ tăng 1 đơn vị.

Code:

Hình : Đếm số lần ma trận E xuất hiện trong ma trận A

* Trả về giá trị biến đếm (count ) khi hoàn thành xong vòng lặp ta sẽ thu được kết quả cuối cùng số lần xuất hiện của E trong A.

Code :

Hình : Đếm số lần ma trận E xuất hiện trong ma trận A

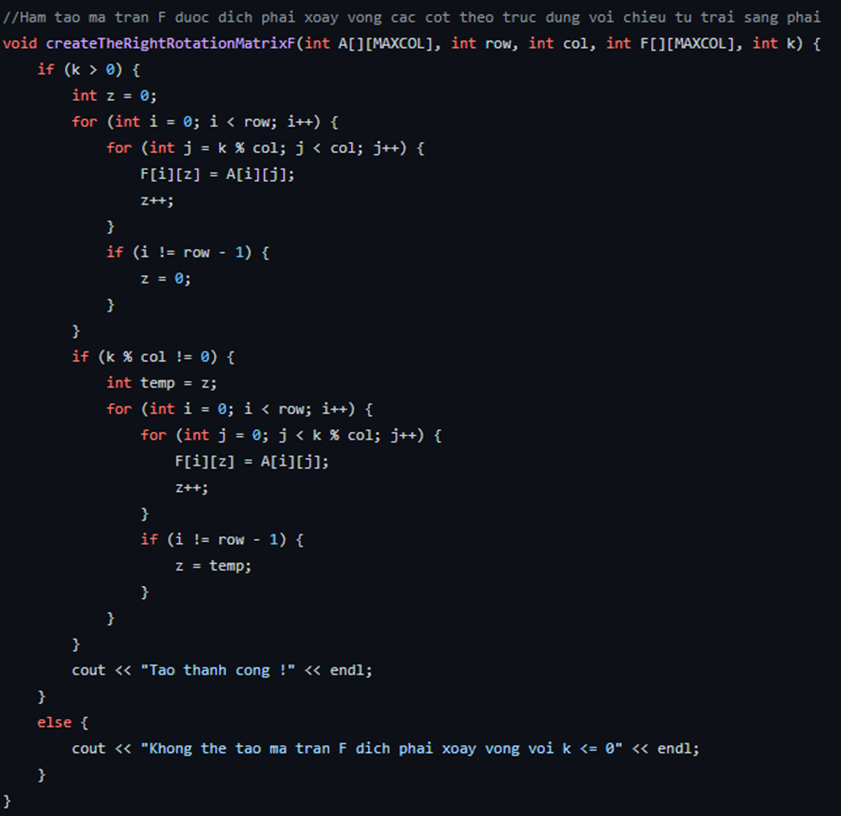
## Từ A đã cho hãy tạo và xuất ra một ma trận F được dịch phải xoay vòng các cột theo trục đứng với chiều từ trái sang phải k lần (k > 0 được nhập từ bàn phím).

Ý tưởng thuật toán :

* Từ ma trận A, tạo 1 ma trận F.
* Nếu k được nhập từ bàn phím lớn hơn 0 :

+ Tạo hàm lồng for với biến i xuất phát từ 0 và j xuất phát từ số dư của k/col(Vì khi k > col mà k chia hết cho col => Ma trận xoay đúng 1 vòng và ma trận F = ma trận A, tương tự những số dư sau và các số dư sẽ không thể lớn hơn Col được ). Tạo thêm 1 biến bên ngoài z = 0 ( đặt tại hàm for của biến i thêm z = 0 để mỗi khi thực hiện xong vòng lặp của j thì z được trả về giá trị 0 để xét tiếp)  và cho tăng dần trong hàm for để thực hiện đổi chỗ từ vị trí này của ma trận A sẽ là vị trí mới trong ma trận F.

* Nếu k được nhập từ bàn phím nhỏ hơn 0 : Xuất thông báo không thể tạo ma trận.

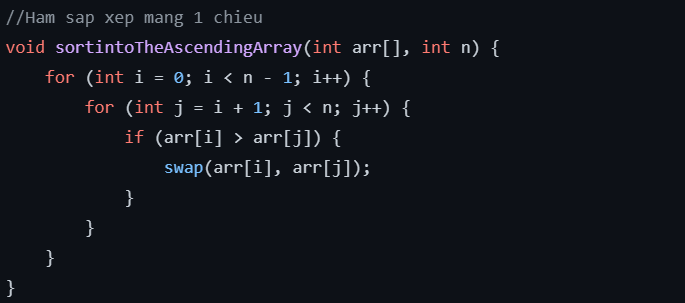
Code :

Hình : Ma Trận F được dịch phải xoay vòng các cột theo trục đứng với chiều từ trái sang phải k lần

## Từ A đã cho hãy tạo và xuất ra một ma trận I được xoắn ốc giảm dần ngược chiều kim đồng hồ.

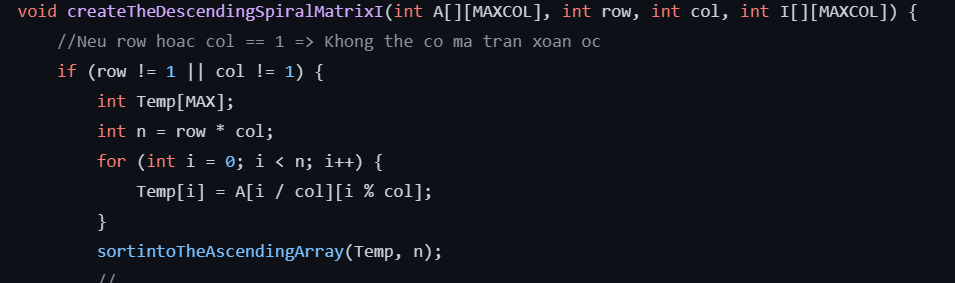
Ý tưởng thuật toán :

* Khởi tạo hàm sắp xếp mảng 1 chiều giảm dần : Cho vòng lặp for kiểm tra các phần tử của mảng A nếu phần tử A[i] > A[j] thì đổi chỗ A[i] và A[j] bằng hàm swap.

Code :

Hình : Xắp xếp mảng 1 chiều

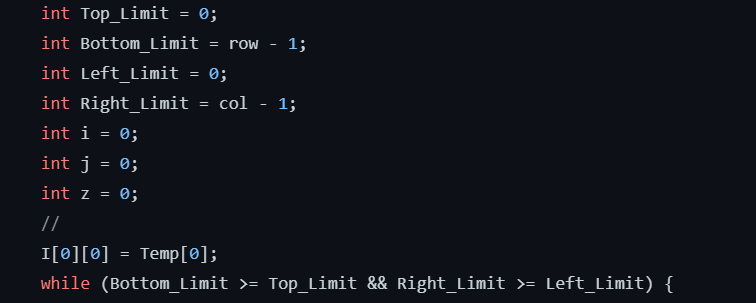
* Tạo hàm xắp xếp ma trận I xoắn ốc giảm dần ngược chiều kim đồng hồ :
* Kiểm tra số dòng số cột của ma trận A nhận vào, kiểm tra số dòng và số cột có khác 1 hay không nếu bằng 1 thì xuất ra màn hình một câu thông báo không thẻ tạo ma trận xoắn ốc ngược lại nếu số dòng và số cột khác 1 thì khởi tạo một biến tạm Temp[MAX] và n bằng số dòng nhân với số cột sau đó sử dụng vòng lặp for để tiến hành đưa giá trị của mảng 1 chiều vào ma trận xoắn ốc.

Code :

Hình 27: Tạo ma trận xoắn ốc

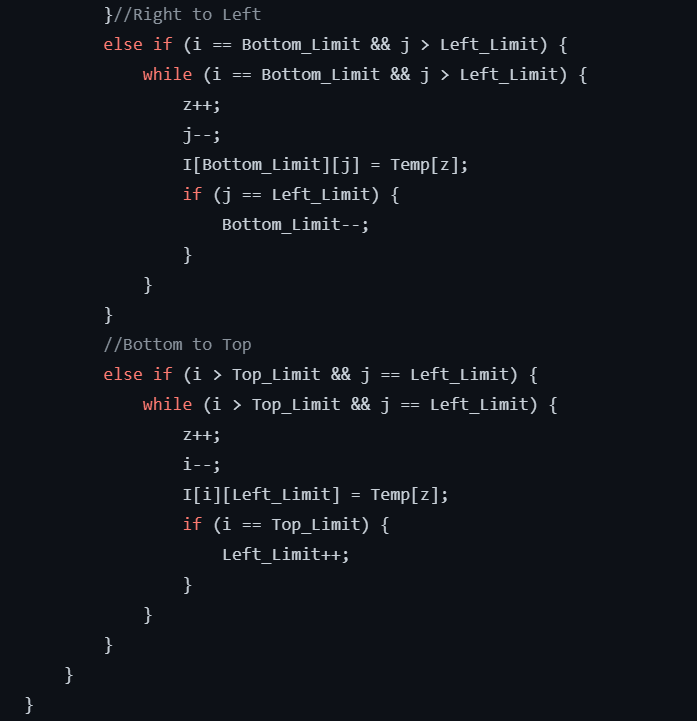
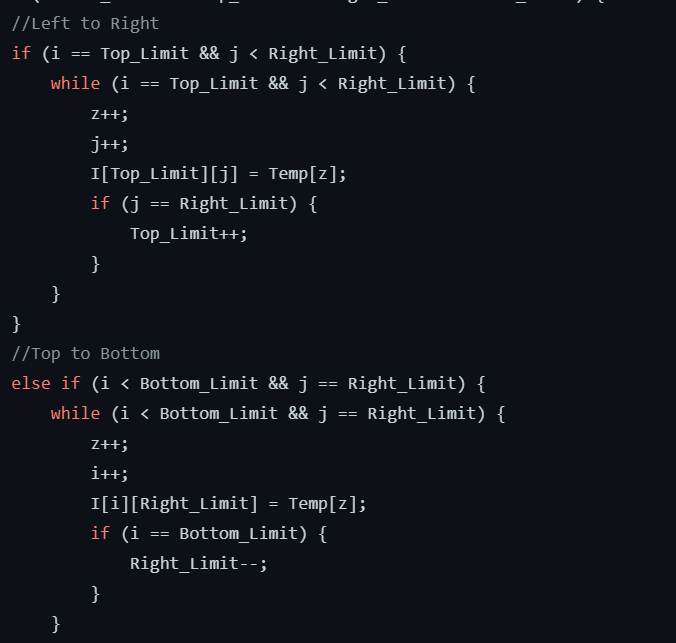
* Xắp xếp mảng 1 chiều thành ma trận xoắn ốc :

+ Đánh số cột đầu tiên và cột cuối cùng theo chiều tăng dần từ trái sang phải, đánh dấu các dòng theo chiều tăng dần từ trên xuống dưới.

Code :

Hình 28: Tạo ma trận xoắn ốc

+ Phân ma trận thành 4 mảnh : từ trái sang phải, từ trên xuống dưới từ phải sang trái và từ dưới lên trên để tiến hành xắp xếp. Sau mỗi lẫn sắp xếp thì bỏ đi dòng và cột đã thao tác cho đến khi Top\_limit tăng Left\_Limit tăng và Bottom\_Limit giảm Right\_Limit giảm cho đến khi Top\_Limit bằng Bottom\_Limit và Left\_limit bằng Right\_Limit thì dừng. Kết thúc ma trận ta có ma trận xoắn ốc

 Code :

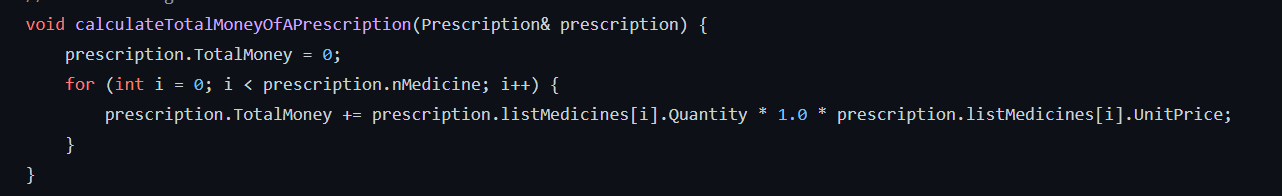
Hình 29: Tạo ma trận xoắn ốc

# Cấu trúc

## Tính tổng tiền của một đơn thuốc.

Ý tưởng thuật toán:

* Tạo hàm tính tổng tiền một đơn thuốc
* Khai báo một biến TotalMoney
* Cho chạy vòng lập for để thêm giá trị vào TotalMoney.

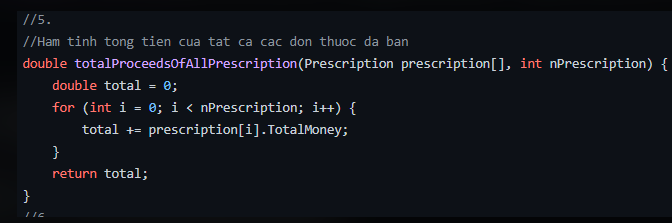
Code:

Hình : Tính tổng tiền một đơn thu

## Tính tổng tiền mà nhà thuốc đã thu được khi bán xong các đơn thuốc.

Ý tưởng thuật toán:

* Tạo hàm tính tổng tiền các đơn thuốc
* Khai báo biến total = 0;
* Tiếp đó cho chạy vòng lập for để cộng tiền bán từng đơn thuốc
* Sau cùng return về total.
* Kết quả: Thu được giá trị sau khi bán tất cả đơn thuốc.

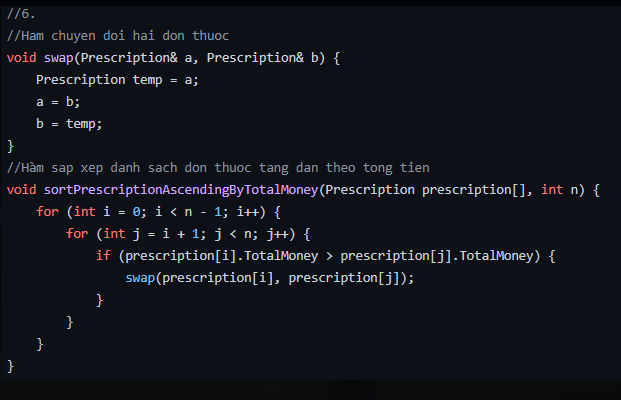
Code:

Hình : Tính tổng tiền mà nhà thuốc đã thu được khi bán xong các đơn thuốc

## Sắp xếp đơn thuốc tăng dần theo tổng tiền.

Ý tưởng thuật toán:

* Bắt đầu với một danh sách các đơn thuốc nhập từ file,
* Tạo hàm swap(a,b).
* Tạo hàm sắp xếp các đơn thuốc
* Cho chạy 2 vòng lập I va J.
* Kiểm tra điều kiện nếu giá trị của tổng tiền thứ nhất lớn hơn tổng tiền thứ 2 thì ta gọi hàm swap( (prescription[i], prescription[j]).
* Kết quả thu được danh sach được sắp xếp tăng dần theo tổng tiền

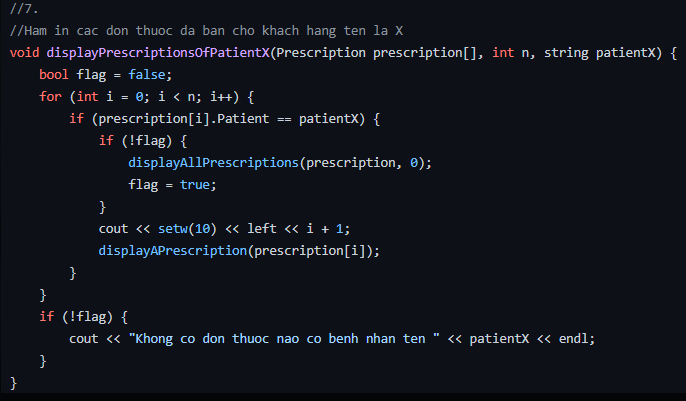
Code:

Hình : Sắp xếp đơn thuốc tăng dần theo tổng tiền.

## Hãy liệt kê các đơn thuốc đã bán cho khách hàng tên là X

Ý tưởng thuật toán:

* Tạo biến flag = false
* Cho chạy vòng lập for
* kiểm tra điều kiện( nếu tên khách hàng == Khachhangx)
* Nếu đúng gọi hàm in đơn thuốc
* Kết quả: thu được đơn thuốc bán cho khách hàng x

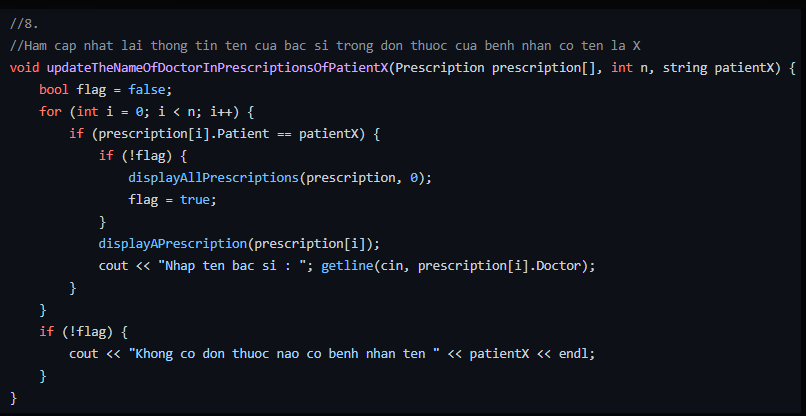
Code:

Hình : Hãy liệt kê các đơn thuốc đã bán cho khách hàng tên là X

## Cập nhật lại thông tin tên bác sĩ trong don thuoc cua benhnhanX

Ý tưởng thuật toán:

* Tạo hàm cập nhật lại thông tin bác sĩ trong đơn thuốc
* Tạo biến flag = false
* Cho chạy vòng lập for
* Kiểm tra điều kiện( nếu tên bệnh nhân == benhnhanx)
* Nếu đúng xuất tiêu đề gọi hàm in ,nhập tên bác sĩ
* Nếu không tìm thấy benhnhanX sẽ in ra thông báo cho người dùng nhận biết
* Kết quả: nhận dược thong tin bác sĩ (đã cập nhật lại tên) của bệnh nhân X.

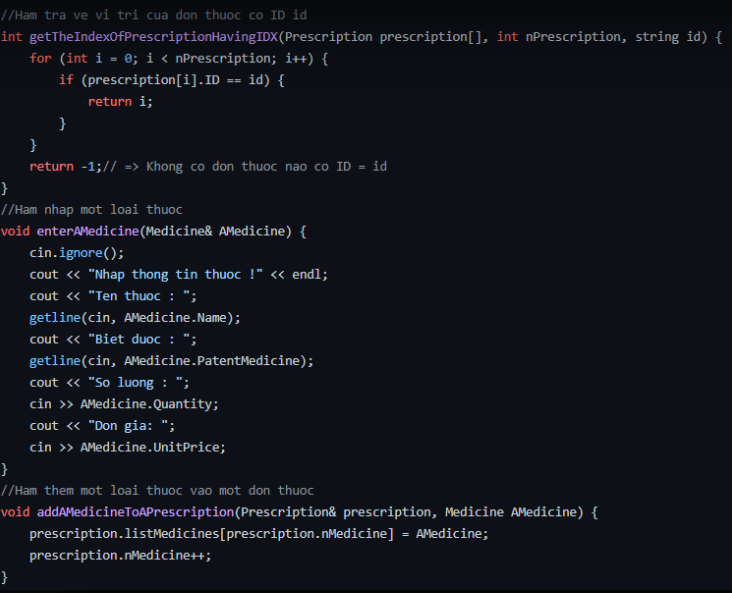
Code:

Hình : Cập nhật lại thông tin tên của bác sĩ trong đơn thuốc của bệnh nhân có tên là X

## Thêm một loại thuốc vào đơn thuốc có mã X trong danh sách.

Ý tưởng thuật toán:

* Nhập vào một ID.
* Tạo hàm trả về vị trí của đơn thuoc có ID trùng với id nhập vào (Nếu tìm thấy trả về vị trí của I. Ngược lại trả về -1).
* Tạo một hàm nhập loại thuốc
* Taọ một hàm để thêm một loại thuốc vào đơn thuốc.
* Tại hàm main ta thực hiện:
* Khai báo k = hàm tìm vị trí của đơn thuốc.
* Kiểm tra điều kiện.Nếu k != -1 thì ta gọi hai hàm nhập và thêm loại thuốc vào trong đơn thuốc tại vị trí k.
* Kết quả: Thu được danh sách các đơn thuốc (Sau khi thêm một loại thuốc vào mã X).

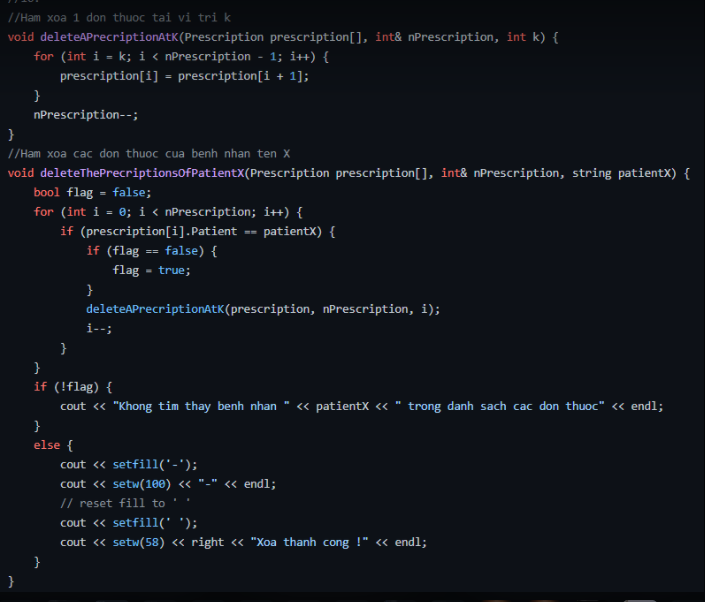
Code:

Hình : Thêm một loại thuốc có mã thuốc là X trong danh sách

## Xóa các đơn thuốc mà bệnh nhân X đã mua.

Ý tưởng thuật toán:

* Tạo hàm xoá đơn thuốc tại vị trí k.
* Tạo hàm xoá đơn thuốc của bệnh nhân X.
* Ta chạy vòng lặp for
* Kiểm tra điều kiện nếu tên bênh nhân == benhnhanX.
* Gọi hàm xoá đơn thuốc tại k
* Giảm biến I đi 1
* Kết quả:thu được danh sách các đơn thuốc sau khi các đơn thuốc bệnh nhân X đã bị xoá.

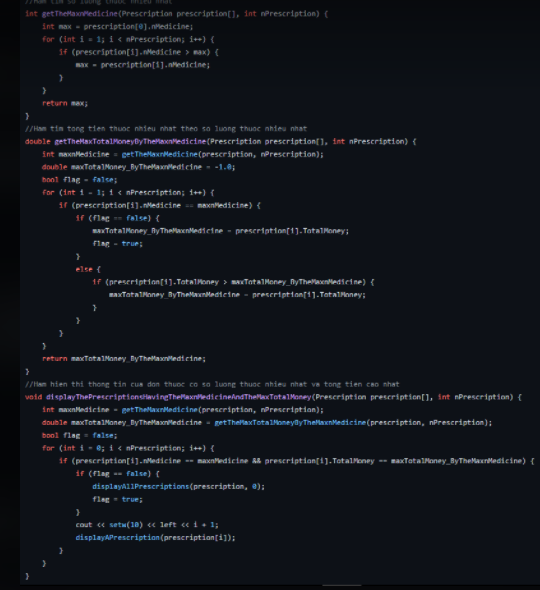
Code:

Hình : Xóa một đơn thuốc mà bênh nhân X đã mua

## Cho biết loại thuốc nào được nhà thuốc bán ra nhiều nhất cùng với số lượng của nó.

## Ý tưởng thuật toán:

* Tạo hàm tìm số lượng thuốc nhiều nhất
* Tạo hàm tìm tổng tiền nhiều nhất và số lượng thuốc nhiều nhât
* Khai báo biến maxmedicine = hàm tìm số lượng thuốc nhiều nhất.
* Khai báo biến tổng tiền lớn nhất = -1.0;
* Khai báo biến flag.
* Cho chạy vòng lập for
* Nếu số lượng của đơn thuốc == maxmedicine.
* If flag == false. Thì tổng tiền lớn nhất =tổng tiền của đơn thuốc[i].trả flag = true.Ngược lại , nếu tổng tiền đơn thuốc [i] lơn hơn tổng tiền lớn nhất thì : Tổng tiền lớn nhất = tổng tiền đơn thuốc [i].
* Return về biến tổng tiền lớn nhất.
* Tạo hàm hiển thị thông tin
* Khai báo maxnMedicine == hàm tìm số lượng thuốc nhiều nhất
* Khai báo maxTotalMoney\_ByTheMaxnMedicine == hàm tìm tổng tiền lớn nhất
* Cho chạy vòng lập for.
* Kiểm tra điều kiện nếu thoả gọi hàm xuất thông tin của đơn thuốc[i].
* Kết quả: thu được thông tin tổng tiền thuốc nhiều nhất và số lượng lớn nhất.

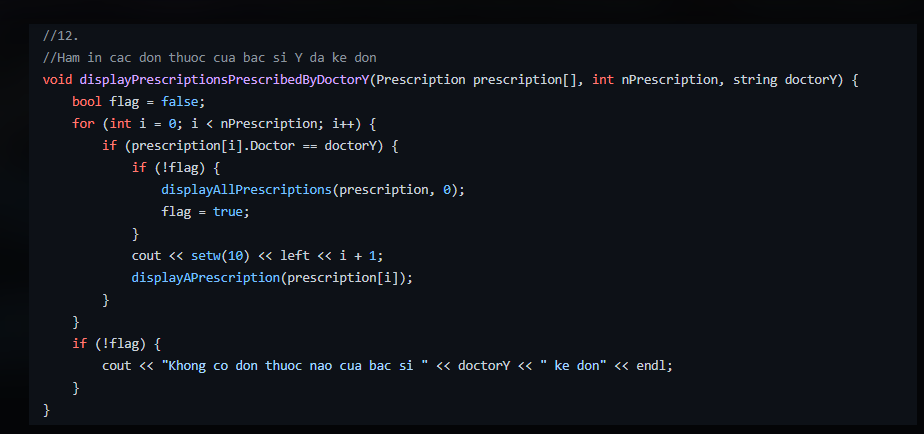
Code:

Hình : Cho biết loại thuốc nào được nhà thuốc bán ra nhiều nhất cùng với số lượng của nó

## Liệt kê các đơn thuốc của bác sĩ Y đã kê đơn.

Ý tưởng thuật toán:

* Tạo hàm in các đơn thuốc bác sĩ y kê đơn.
* Tạo biến flag = false
* Cho chạy vòng lập for
* Kiểm tra điều kiện( nếu tên bác sĩ == bacsiY).Nếu đúng gọi hàm in đơn thuốc.
* Kết quả thu được đơn thuốc bác sĩ y kê đơn.

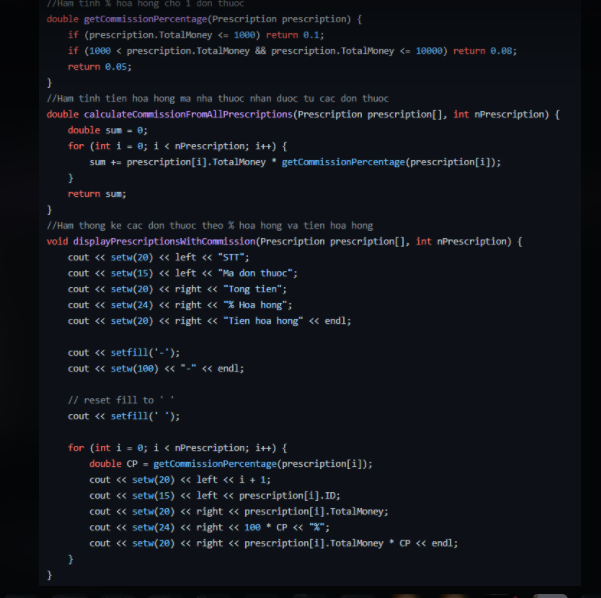
Code:

Hình : Liệt kê các đơn thuốc mà bác sĩ Y đã kê đơn

## Tính hoa hồng mà tiệm thuốc nhận được khi bán các đơn thuốc. Biết rằng tiền hoa hồng của các hóa đơn được xác định dựa vào tổng tiền như sau: Nếu tổng tiền nhỏ hơn 1000 thì hoa hồng là 10%, nếu 1000<tổng tiền<=10000 thì hoa hồng là 8%, còn lại hoa hồng là 5%..

Ý tưởng thuật toán:

* Tạo hàm tính hoa hồng cho đơn thuốc
* Nếu tổng tiền <= 1000 trả về 0.1.1000<tổng tiền <10000 trả về 0.08. tổng tiền>10000 trả về 0.05.
* Tạo hàm tính tiền hoa hồng mà nhà thuốc nhận được sau khi bán các đơn thuốc
* Khai báo biến sum =0;
* Chạy vòng lập for sau đó sum += tổng tiền \* tiền hoa hồng (hàm trên)
* Cuối cùng return về biến sum.
* Tạo hàm thống kê các đơn thuốc theo % hoa hồng và tiền hoa hồng
* Xuát tiêu đề
* Chạy vòng lâp for để in ra các phần tử.
* Kết quả: Thu dược danh sách các đơn thuốc bao gồm cả phần trăm hoa hồng và tiền hoa hồng.

Code:

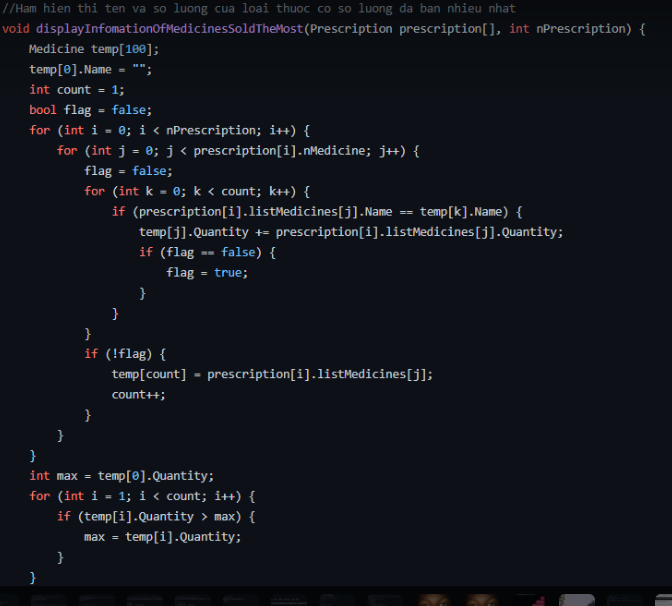
Hình : Tính tiền hoa hồng của tiệm thuốc

## Cho biết loại thuốc nào được nhà thuốc bán ra nhiều nhất cùng với số lượng của nó.

Ý tưởng thuật toán:

* Tạo hàm hiển thị tên và số lượng của loại thuốc bán nhiều nhất
* Khởi tạo Medicine temp[100].
* Khai báo temp[0].name = “ ”.
* Khai báo biến đếm count – 1.
* Chạy 3 vòng lập I j k .
* Sau đó kiểm tra điều kiện :(prescription[i].listMedicines[j].Name == temp[k].Name).
* Nếu dúng biến temp[j].Quatity += số lượng của loại thuốc.
* Bên ngoài vòng lập k tiếp tục kiểm tra điều kiện (!flag).Nếu thoả điều kiện biến temp[count] = prescription[i].listMedicines[j].
* Tăng biến count lên.
* Tạo một vòng lập I để tìm max.
* Cuối cùng dùng vòng lập for và xét điều kiện nếu tìm thấy:
* Xuất ra Tên cùng với Số lượng của loại thuốc.
* Kết quả : Thu được Tên và Số lượng của loại thuốc đã được nhà thuốc bán ra nhiều nhất.

Code:

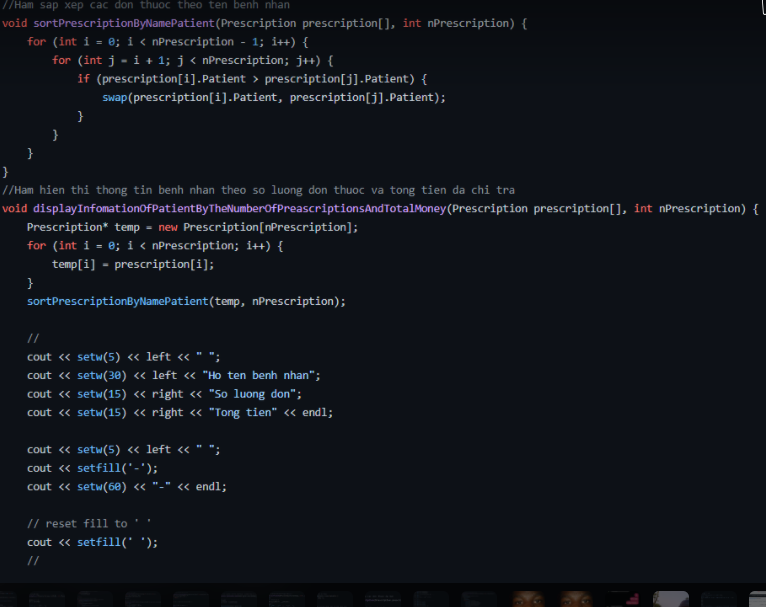
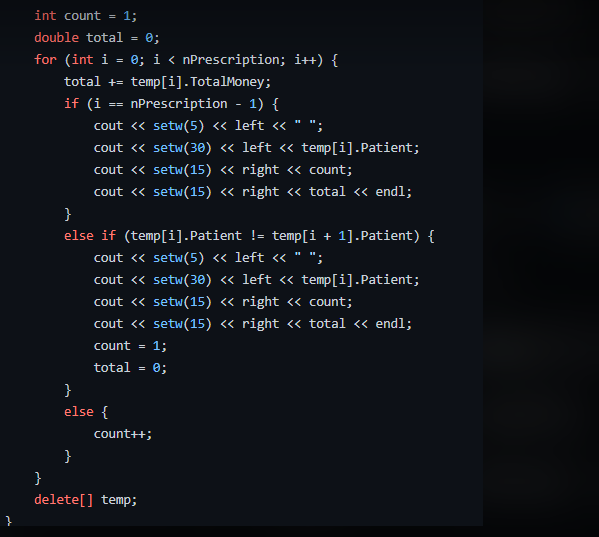


Hình : Cho biết loại thuốc nào được bán ra nhiều nhất và số lượng của nó

## Thống kê theo bệnh nhân số tiền và số đơn thuôc họ đã mua tại nhà thuốc theo mẫu.

Ý tưởng thuật toán:

* Tạo hàm sắp xếp các đơn thuốc theo bệnh nhân,
* Chạy 2 vòng lập I,j .
* Kiểm tra điều kiện nếu thoả ta gọi hàm swap.
* Tạo hàm hiển thị thông tin bệnh nhân theo yêu cầu
* Khởi tạo biến Prescription\* temp = new Prescription[nPrescription];
* Dùng vòng lập for để truyền cho biến temp.
* Gọi hàm sắp xếp các đơn thuốc vừa tạo bên trên.
* Chạy vòng lập I kiểm tra điều kiện nếu (i == nPrescription – 1 ) hoặc (temp[i].Patient != temp[i + 1].Patient) thì in ra danh sách. Ngược lại tăng biến count lên dần .
* Xoá Mảng.
* Kết quả : In ra danh sách bệnh nhân theo yêu cầu đề bài.

Code:

Hình : Thống kê bệnh nhân và đơn thuốc của bệnh nhân đã mua

# Tổng kết - Đánh giá công việc nhóm:

## Công việc cá nhân:

* Mỗi thành viên đều làm tốt 100% bài tập của mình được phân công thực hiện tương đương với 20% tổng số bài tập trong nhóm.

## Đánh giá chung:

* Mặc dù còn kha khá sai sót nhưng tất cả đều rất chăm làm, tìm đủ mọi cách để hoàn thành bài tập của mình.
* Các thành viên đều thực hiện bài tập và tự giác gửi đúng hẹn không chậm trễ gây ảnh hưởng đến công việc chung của nhóm.
* Khi họp nhóm các thành viên đều có mặt đầy đủ và đưa ra những ý kiến để cải thiện bài tập.
* Mọi người đều lắng nghe ý kiến của nhau và tự nhận ra sai lầm của mình rồi sửa chữa ngay.