**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



**BÁO CÁO ĐỒ ÁN 1**

**TÍNH TOÁN TRÊN**

**MA TRẬN**

**SVTH1 : NGUYỄN THÀNH TÂM - 17110219**

**SVTH2 : LÊ KIM ĐỈNH - 17110123**

**GVPT: GV TỪ THỊ TUYẾT HỒNG**

**TP. Hồ Chí Minh, tháng 11 năm 2019**

MỤC LỤC

[LỜI MỞ ĐẦU 1](#_Toc24232017)

[Danh mục các hình 2](#_Toc24232018)

[Danh mục các bảng 4](#_Toc24232019)

[CHƯƠNG 1 : TỔNG QUAN ĐỒ ÁN 5](#_Toc24232020)

[1.1. Lịch sử hình thành 5](#_Toc24232021)

[1.2. Giới thiệu về ma trận 5](#_Toc24232022)

[1.3. Lý do, mục đích, mục tiêu chọn đề tài 5](#_Toc24232023)

[1.3.1. Lý do, mục đích chọn đề tài 5](#_Toc24232024)

[1.3.2. Mục tiêu chọn đề tài 6](#_Toc24232025)

[1.4. Lựa chọn ngôn ngữ cài đặt mô phỏng 6](#_Toc24232026)

[CHƯƠNG 2 : NỘI DUNG 7](#_Toc24232027)

[2.1. Quá trình và công việc thực hiện đồ án 7](#_Toc24232028)

[2.2. Khái quát thuật toán 7](#_Toc24232029)

[2.2.1. Phép toán cộng (trừ) hai ma trận 7](#_Toc24232030)

[2.2.2. Phép toán nhân hai ma trận 7](#_Toc24232031)

[2.2.2. Chuyển vị ma trận 7](#_Toc24232032)

[2.2.3. Nghịch đảo ma trận 7](#_Toc24232033)

[2.2.2. Ứng dụng 8](#_Toc24232034)

[2.3. Thiết kế giao diện 8](#_Toc24232035)

[2.5. Cách cài đặt và viết chương trình 8](#_Toc24232036)

[2.5.1. Một số hàm xử lý chức năng 8](#_Toc24232037)

[2.5.2. Giải thích code 10](#_Toc24232038)

[2.6 Sản phẩm của đồ án 23](#_Toc24232039)

[CHƯƠNG 3 : KẾT LUẬN 24](#_Toc24232040)

[3.1 Kết luận 24](#_Toc24232041)

[3.2 Ưu điểm 24](#_Toc24232042)

[3.3 Nhược điểm 24](#_Toc24232043)

[3.4 Đề xuất phương án cải thiện 24](#_Toc24232044)

[PHỤ LỤC 25](#_Toc24232045)

[TÀI LIỆU THAM KHẢO 26](#_Toc24232046)

LỜI MỞ ĐẦU

Nhóm thực hiện xin chân thành cảm ơn sự hướng dẫn tận tình của cô Từ Thị Tuyết Hồng, cả về chuyên môn lẫn định hướng cho nhóm. Vì kiến thức còn hạn hẹp và thời gian có hạn nên nhóm chưa thể tối ưu hết các thuật toán sử dụng trong ứng dụng cũng như không thể tránh khỏi các thiếu sót. Vì vậy rất mong nhận được sự góp ý của cô để đồ án của nhóm có thể hoàn thiện hơn.

Danh mục các hình

[Figure 1: Phân công công việc 7](#_Toc24311821)

[Figure 2 Công thức nhân ma trận 7](#_Toc24311822)

[Figure 3: Giao diện chính 8](file:///C:\Users\mrtht\Downloads\BAOCAODOANHK11819-1.docx#_Toc24311823)

[Figure 4 Thuộc tính, hàm dựng, properties 9](#_Toc24311824)

[Figure 5 Nhập xuất 9](#_Toc24311825)

[Figure 6 In ra màng hình 11](#_Toc24311826)

[Figure 7 In ra file TXT 11](#_Toc24311827)

[Figure 8 In ra file Excel 12](#_Toc24311828)

[Figure 9 In ra file Excel 12](#_Toc24311829)

[Figure 10 RandomMatrix 13](#_Toc24311830)

[Figure 11 Nhập ma trận bằng tay 13](#_Toc24311831)

[Figure 12 Nhập ma trận bằng tay 14](#_Toc24311832)

[Figure 13 Đọc ma trận từ file TXT 14](#_Toc24311833)

[Figure 14 Cộng ma trận 15](#_Toc24311834)

[Figure 15 Trừ ma trận 16](#_Toc24311835)

[Figure 16 Nhân ma trận 16](#_Toc24311836)

[Figure 17 Công thức nhân ma trận 17](#_Toc24311837)

[Figure 18 Chuyển vị ma trận 17](#_Toc24311838)

[Figure 19 Lấy độ dài 18](#_Toc24311839)

[Figure 20 Công thức tính độ dài 18](#_Toc24311840)

[Figure 21 Chuẩn hóa ma trận 18](#_Toc24311841)

[Figure 22 Độ tương đồng cosin giữa các vecto trogn ma trận 19](#_Toc24311842)

[Figure 23 Ví dụ tính độ tương đối 19](#_Toc24311843)

[Figure 24 Ví dụ tính độ tương đối 20](#_Toc24311844)

[Figure 25 Ví dụ tính độ tương đối 20](#_Toc24311845)

[Figure 26 Ma trận nghịch đảo 20](#_Toc24311846)

[Figure 27 Ví dụ nghịch đảo 21](#_Toc24311847)

[Figure 28 Tìm ma trận con 21](#_Toc24311848)

[Figure 29 Tính định thức 22](#_Toc24311849)

[Figure 30 Ma trận 23](#_Toc24311850)

Danh mục các bảng

[Bảng 1: Bảng phân công 23](#_Toc531297648)

CHƯƠNG 1 : TỔNG QUAN ĐỒ ÁN

1.1. Lịch sử hình thành

Ma trận có một lịch sử ứng dụng lâu dài trong việc giải các phương trình tuyến tính nhưng chúng được gọi là mảng cho đến những năm 1800. Văn bản *Trung Quốc Chín chương* về nghệ thuật toán học được viết vào thế kỷ thứ 10 - 2 trước Công nguyên là ví dụ đầu tiên về việc sử dụng các phương pháp mảng để giải các phương trình đồng thời, bao gồm cả khái niệm các định thức. Năm 1545, nhà toán học người Ý Gerolamo Cardano đã mang phương pháp này đến châu Âu khi ông xuất bản *Ars Magna*. Nhà toán học Nhật Bản Seki đã sử dụng các phương pháp mảng tương tự để giải các phương trình đồng thời vào năm 1683. Nhà toán học người Hà Lan Jan de Witt đã trình bày các phép biến đổi bằng cách sử dụng các mảng trong cuốn sách *1659* *Element of Curves (1659).* Từ năm 1700 đến 1710, Gottfried Wilhelm Leibniz đã công khai việc sử dụng mảng để ghi thông tin hoặc giải pháp và thử nghiệm với hơn 50 hệ thống mảng khác nhau. Cramer trình bày quy tắc của mình vào năm 1750.[1]

1.2. Giới thiệu về ma trận

Ma trận là một mảng hình chữ nhật của các số hoặc các đối tượng toán học khác mà các hoạt động như cộng và nhân được xác định. Thông thường nhất, một ma trận trên trường F là một mảng vô hướng hình chữ nhật mà mỗi trong số đó là thành viên của F.

Kích thước của ma trận được xác định bởi số lượng hàng và cột mà nó chứa. Một ma trận có m hàng và n cột được gọi là ma trận m × n hoặc ma trận m-by-n, trong khi m và n được gọi là kích thước của nó.

1.3. Lý do, mục đích, mục tiêu chọn đề tài

1.3.1. Lý do, mục đích chọn đề tài

Sau khi tìm hiểu và nghiên cứu, nhóm quyết định chọn chủ đề “Tính toán trên ma trận”, vì nhóm nhận ra rằng chủ đề này rèn luyện cho nhóm khả năng lập trình, tư duy thuật toán khá tốt. Và điều đặc biệt nhóm nhận ra rằng đại số tuyến tính là một nền tảng rất lớn cho các môn học sau này như machine learning, deep learning và được ứng dụng rộng rãi trong nhiều lĩnh vực, đặc biệt là trong trí tuệ nhân tạo.

Mặc dù chủ đề của đồ án này dành cho ba thành viên, nhưng chúng em muốn thử sức mình để có thêm nhiều kiến thức cũng như khả năng làm việc nhóm hơn, nếu có điều gì sai sót, mong cô góp ý và giúp đỡ.

1.3.2. Mục tiêu chọn đề tài

Với mục tiêu tạo ra ứng dụng ma trận có đầy đủ chức năng sử dụng như cộng, trừ, nhân, hoán vị, nghịch đảo, độ tương đối giữa cosin và các vecto trong ma trận… và tạo ra các phương thức nhập, xuất ma trận từ file Excel và file Txt giúp cho việc nhập xuất dữ liệu dễ dàng hơn cho người dùng.

1.4. Lựa chọn ngôn ngữ cài đặt mô phỏng

Vì chương trình mô phỏng của nhóm biễu diễn thuật toán dưới dạng các đối tượng thuộc tính, hành vi rõ ràng nên nhóm quyết định sử dụng ngôn ngữ Microsoft C#.NET với phiên bản .NET Framework 4.5.2; IDE và compiler nằm trong bộ Microsoft Visual Studio 2019 để cài dặt mô phỏng này.

CHƯƠNG 2 : NỘI DUNG

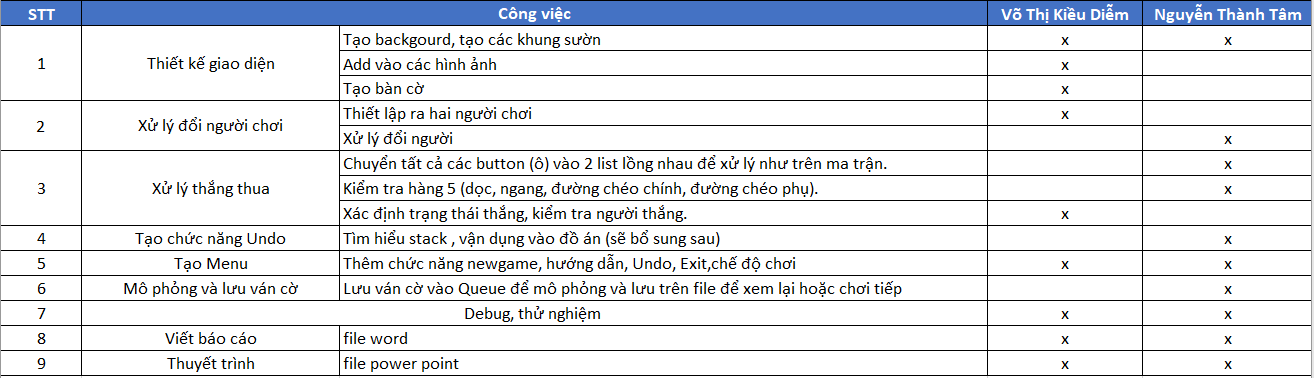
2.1. Quá trình và công việc thực hiện đồ án

Figure 1: Phân công công việc

2.2. Khái quát thuật toán

2.2.1. Phép toán cộng (trừ) hai ma trận

*Tổng* **A**+**B** của hai ma trận cùng kích thước *m*-x-*n* **A** và **B** được một ma trận cùng kích thước với phần tử trong vị trí tương ứng bằng tổng của hai phần tử tương ứng của mỗi ma trận: (**A** + **B**)*i*,*j* = **A***i*,*j* + **B***i*,*j*, với 1 ≤ *i* ≤ *m* và 1 ≤ *j* ≤ *n*.

2.2.2. Phép toán nhân hai ma trận

Phép nhân hai ma trận được xác định khi và chỉ khi số cột của ma trận bên trái bằng số hàng của ma trận bên phải. Nếu A là một ma trận m-x-n và B là một ma trận n-x-p, thì ma trận tích AB là ma trận m-x-p với các phần tử được xác định theo [tích vô hướng](https://vi.wikipedia.org/wiki/T%C3%ADch_v%C3%B4_h%C6%B0%E1%BB%9Bng) của hàng tương ứng trong A với cột tương ứng trong B:

với 1 ≤ *i* ≤ *m* và 1 ≤ *j* ≤ *p.*

Figure 2 Công thức nhân ma trận

2.2.2. Chuyển vị ma trận

Chuyển vị của ma trận m-x-n A là ma trận n-x-m AT (cũng còn ký hiệu là Atr hay tA) tạo ra bằng cách chuyển hàng thành cột và cột thành hàng:(AT)i,j = Aj,i.

2.2.3. Nghịch đảo ma trận

Ma trận A vuông cấp n được gọi là khả nghịch trên vành *V* nếu tồn tại ma trận A' cùng cấp n sao cho A\*A' = A'\*A = E. Khi đó A' được gọi là ma trận nghịch đảo của ma trận A, ký hiệu là A−1.[2]

2.2.2. Ứng dụng

Ma trận được ứng dụng rộng trong nhiều lĩnh vực như: lý thuyết đồ thị, giải tích và hình học, lý thuyết xác suất và thống kê, đối xứng và các biến đổi trong vật lý học,…

2.3. Thiết kế giao diện

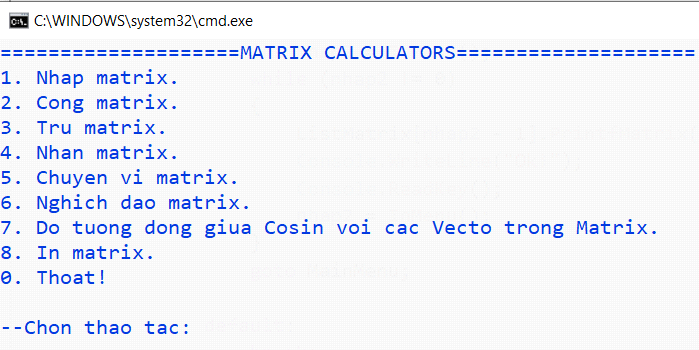


Figure 3: Giao diện chính

Giao diện console đơn giản dễ sử dụng cho phép người dụng theo các thao tác muốn thực hiện như: nhập, cộng, trừ,…

2.5. Cách cài đặt và viết chương trình

2.5.1. Một số hàm xử lý chức năng

Chương trình gồm hai class:

Program.cs: class chính của chương trình, chứa các phương thức liên quan đến xử lý giao diện người dùng.

ClsMaTran.cs: chứa tất cả các thuộc tính và phương thức của ma trận.

ClsMaTran.cs được chia thành các region là:

Các thuộc tính, hàm dựng, properties:

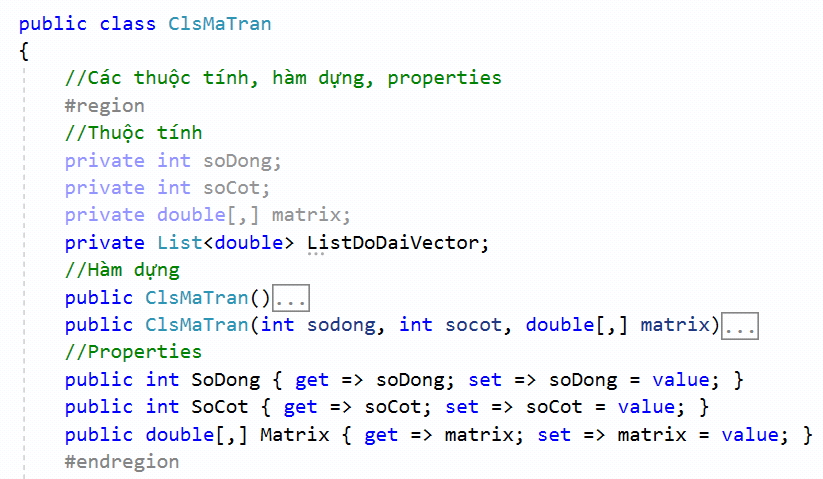


Figure 4 Thuộc tính, hàm dựng, properties

Các thuộc tính gồm: số dòng, số cột của ma trận, và một ma trận kiểm double. Kiểu dữ liệu nhóm chọn là double, vì một số tính toán yêu cầu dộ chính xác cao và sử dụng số thập phân như tìm định thức, tính tương đồng cosin…

Các phương thức nhập xuất:

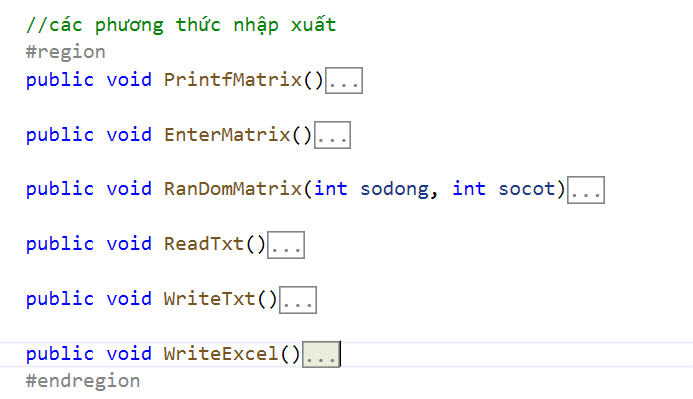


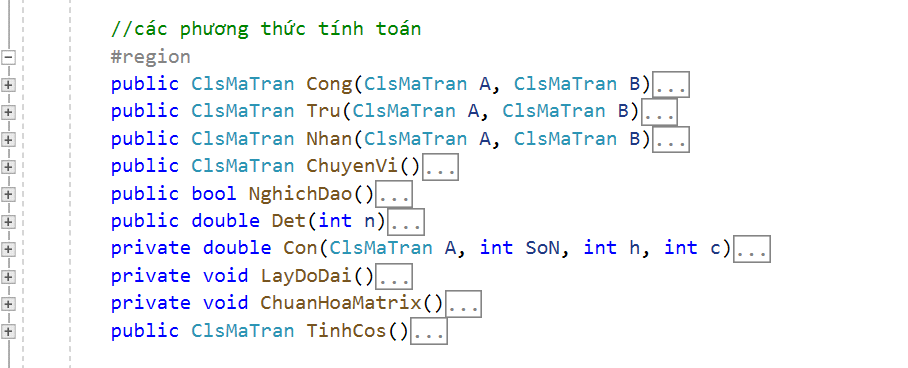
Figure 5 Nhập xuất

Chương trình có đủ các cách nhập xuất ma trận thường dùng, giúp việc nhập xuất trở nên dễ dàng hơn.

Người dùng có thể thực hiện việc nhập từ bàn phím, từ file Txt hoặc file Excel, hoặc người dùng có thể tạo ra ma trận với giá trị ngẫu nhiên. Dữ liệu có thể được xuất ra màn hình, ghi lên file txt và file Excel.

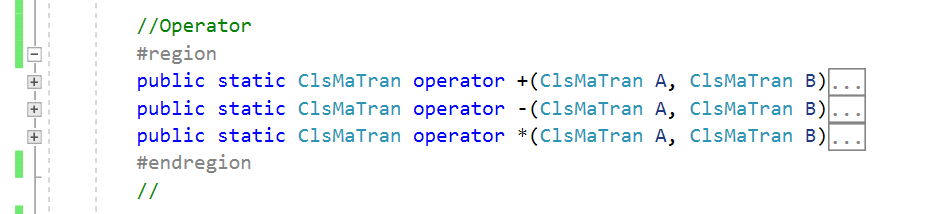
Vì giới hạn số ký tự trên một dòng và giới hạn số dòng trên file Excel nên trường hợp ghi và đọc ma trận có kích thước cực lớn lên file theo cách thông thường sẽ không còn chính xác. Vì vậy các ma trận kích thước lớn nên được ghi lên file Txt theo cách mỗi phần tử trên một dòng.

Các phương thức tính toán:



Region này chứa tất cả các phương thức liên quan đến tính toán ma trận, bao gồm: cộng, trừ, nhân, hoán vị, nghịch đảo, định thức, tương đồng cosin, tìm độ dài vector, chuẩn hóa ma trận.

Các Opertator:



Việc xây dựng các operator giúp người dùng có thể thực hiện các tính toán cơ bản như cộng, trừ, nhân trở nên dễ dàng hơn.

2.5.2. Giải thích code

Các phương thức nhập xuất.

Phương thức PrintMatrix: In ma trận ra màn hình.

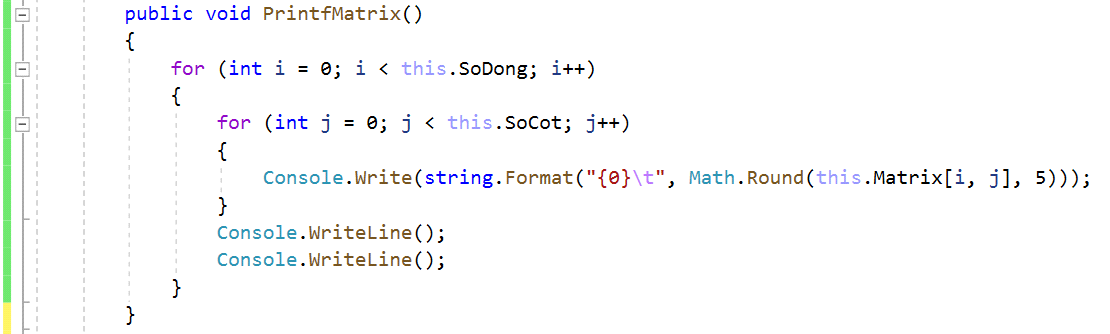


Figure 6 In ra màng hình

Phương thức WriteTxt: ghi ma trận ra file txt.

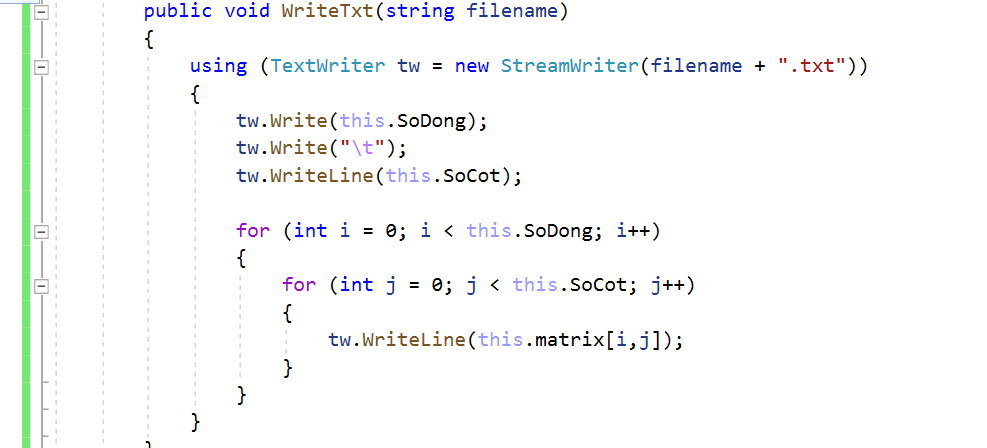


Figure 7 In ra file TXT

Tham số truyền vào là tên file muốn tạo.

Hàng đầu tiên ghi số dòng và số cột, các dòng còn lại, mỗi dòng ghi một phần tử của ma trận.

Phương thức WriteExcel: ghi ma trận lên file Excel.

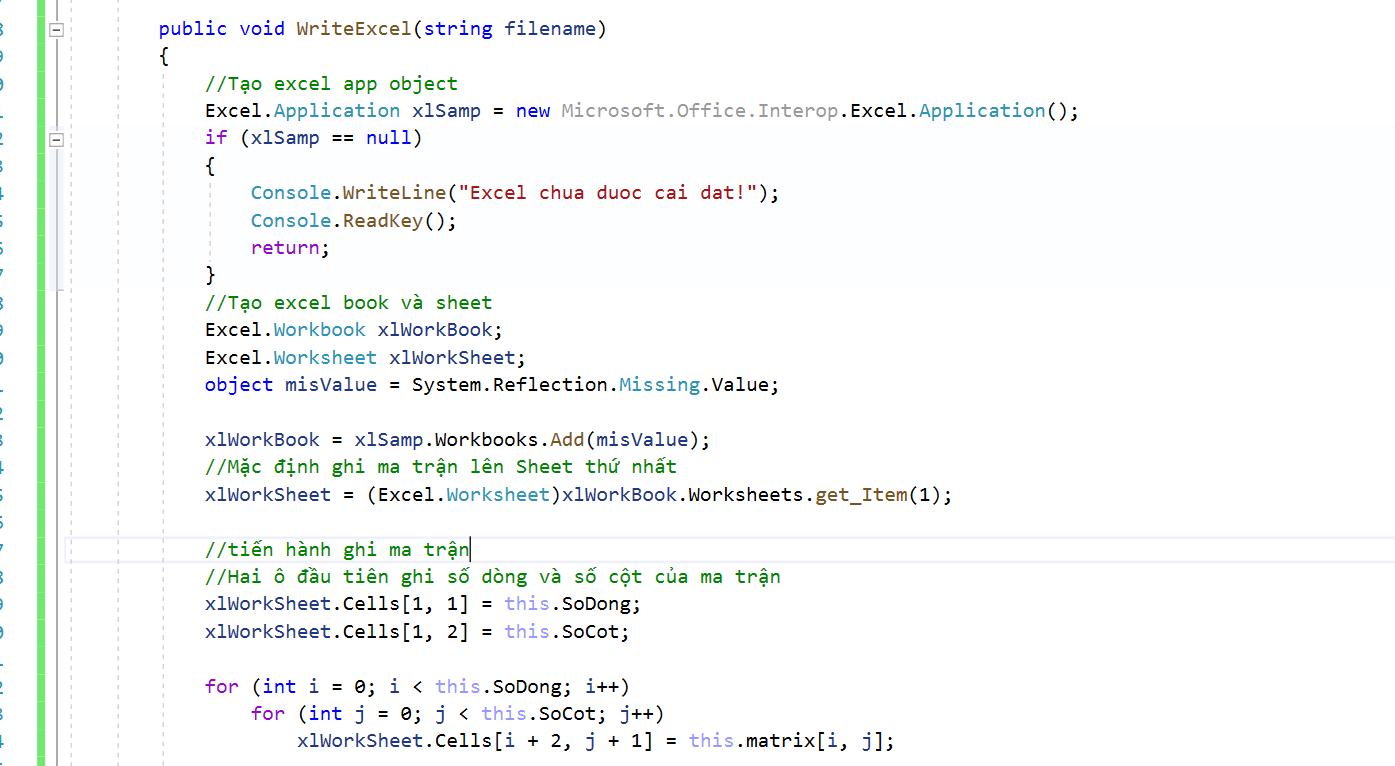


Figure 8 In ra file Excel

Tạo một excel application, nếu không tạo được tức là máy không có sẵn Excel nên sẽ báo lỗi và kết thúc chương trình.

Tạo excel book và sheet, mặc định sẽ ghi ở sheet thứ nhất.

Hai ô đầu tiên sẽ ghi số dòng và số cột của ma trận, các dòng còn lại, mỗi dòng còn lại tương ứng với mỗi dòng của ma trận.

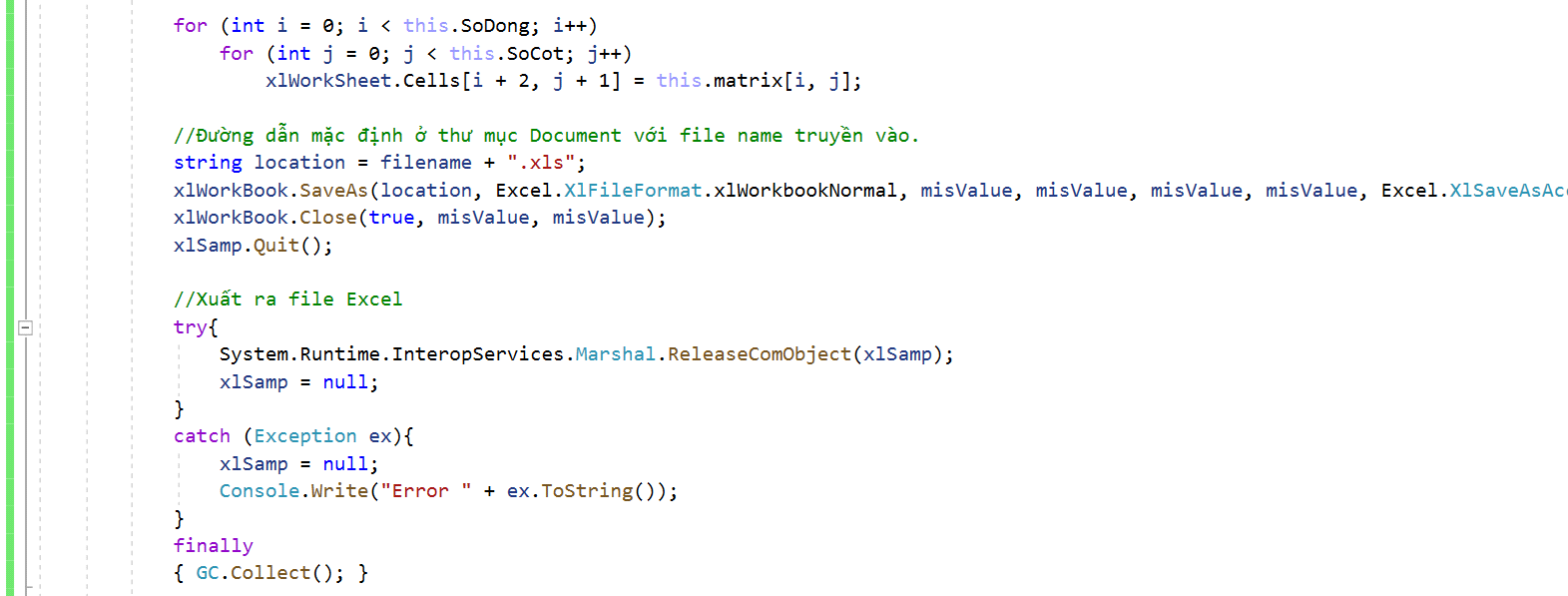


Figure 9 In ra file Excel

String location lưu đường dẫn của file sau khi lưu, có thể thay đổi, và mặc định sẽ lưu ở thư mục Documents.

Sau đó đóng Excel App.

Cuối cùng là xuất ra file excel hoàn chỉnh, nếu có lỗi thì sẽ in lỗi ra console, sau đó là collect memory.

Phương thức RandomMatrix: tạo ma trận có giá trị ngẫu nhiên.

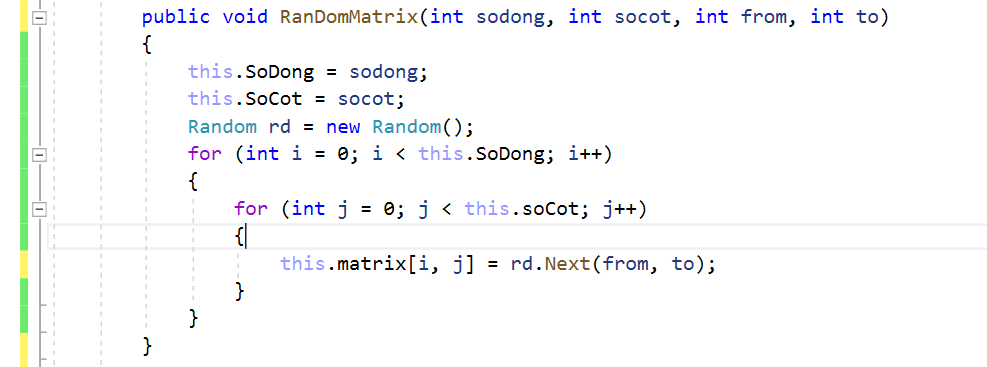


Figure 10 RandomMatrix

Tham số truyền vào bao gồm số dòng, số cột, và khoảng giá trị muốn random.

Phương thức EnterMatrix: tạo ma trận bằng tay.

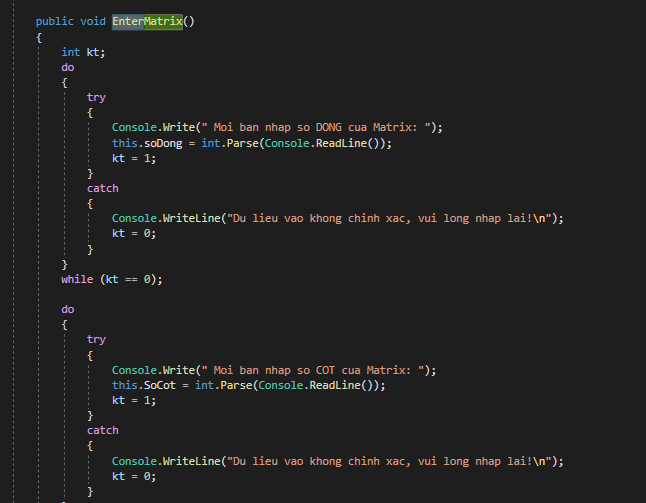


Figure 11 Nhập ma trận bằng tay

Người dùng phải nhập số dòng và số cột trong lúc nhập thì sẽ được kiểm tra nếu người dùng nhập sai bắt nhập lại.

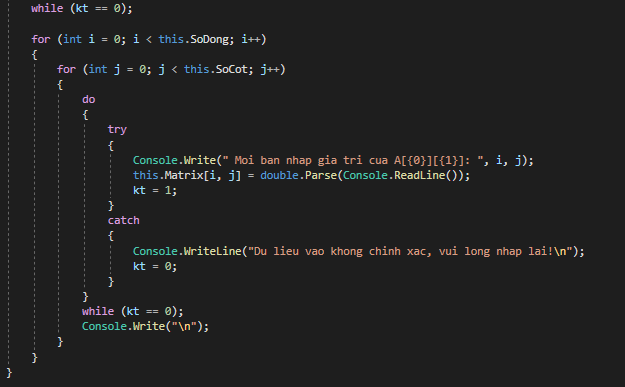


Figure 12 Nhập ma trận bằng tay

Sau khi nhập xong số dòng và số cột thì người dung phải nhập vào giá trị mỗi phần tử.

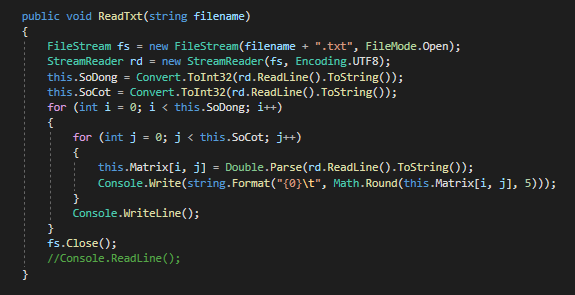


Figure 13 Đọc ma trận từ file TXT

Ban đầu sẽ mở file TXT từ trong máy lên (file này nằm trong project), hai giá trị đầu tương ứng là số dòng và số cột, còn lại là giá trị của ma trận, kết thúc dóng file.

Các phương thức tính toán.

Phương thức cộng ma trận.

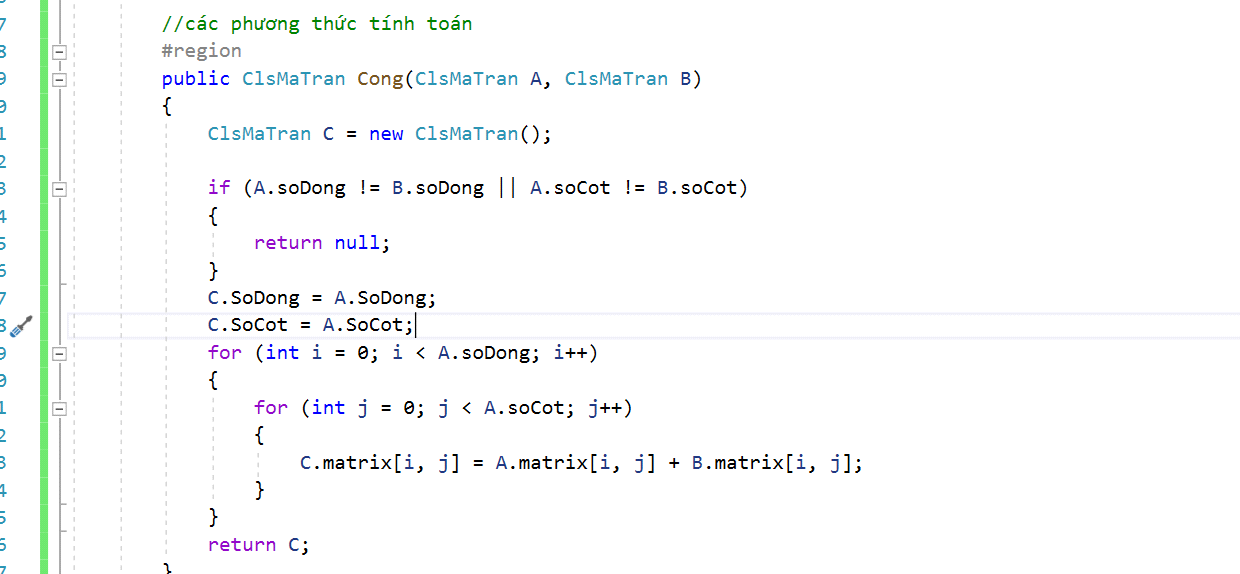


Figure 14 Cộng ma trận

Phương thức truyền vào hai ma trận và trả về một ma trận.

Kiểm tra nếu số dòng dòng và cột của hai ma trận, nếu không bằng nhau thì không cộng được, nên sẽ trả về giá trị null.

Gán số dòng và cột của ma trận trả về bằng với ma trận truyền vào, sau đó dùng hai vòng lặp để cộng từng vị trí tương ứng giữa hai ma trận lại và lưu vào ma trận trả về.

Phương thức trừ ma trận.

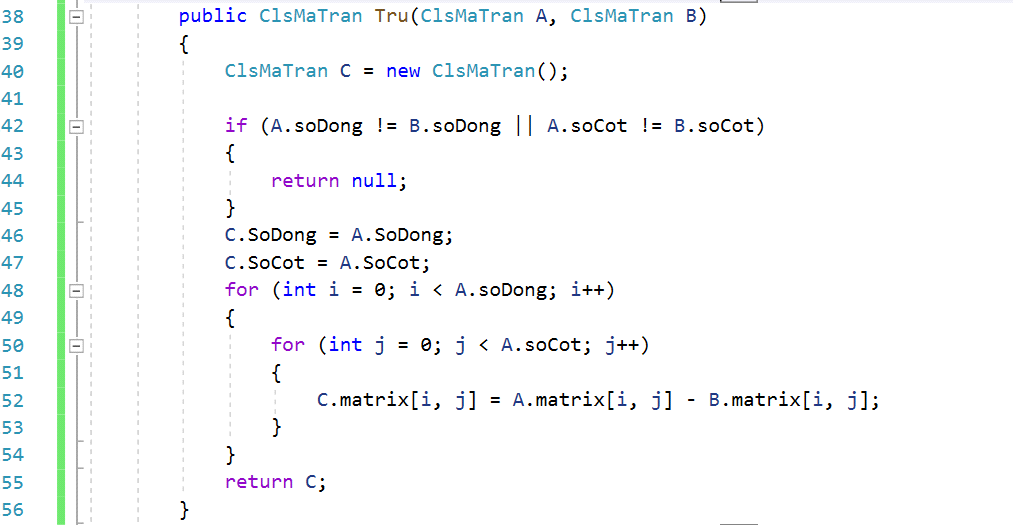


Figure 15 Trừ ma trận

Tương tự như phương thức cộng ma trận.

Phương thức nhân ma trận.

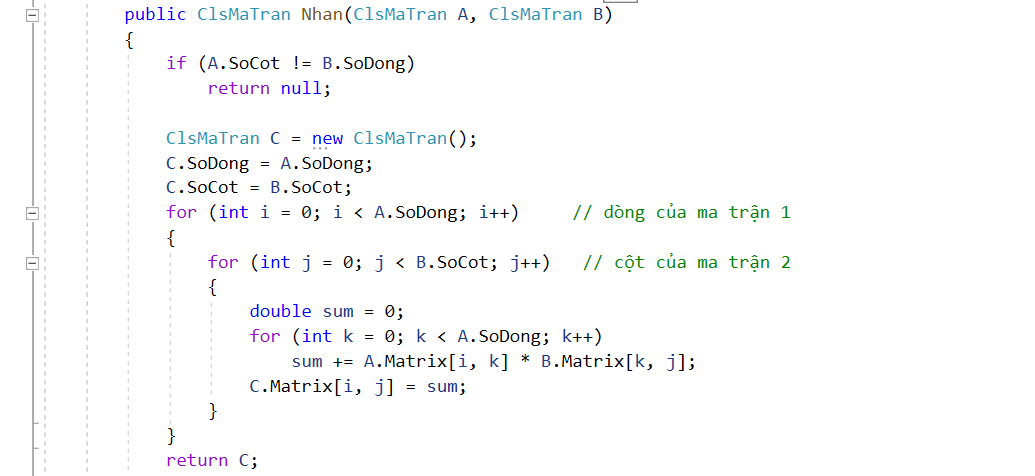


Figure 16 Nhân ma trận

Trước tiên cần kiểm tra điều kiện để nhân hai ma trận, đó là số cột ma trận A phải bằng số cột của ma trận B, nếu không thỏa điều kiện thì trả về Null.

Kết quả trả về của phép nhân ma trận là một ma trận với số dòng bằng với số dòng của ma trận A và số cột bằng số cột ma trận B.

Mỗi phần tử của ma trận kết quả được tính bằng cách lấy hàng của ma trận A nhân với cột của ma trận B, sau đó tính tổng các tích vừa tính được chính là giá trị của ma trận kết quả.

Công thức tính tích hai ma trận:

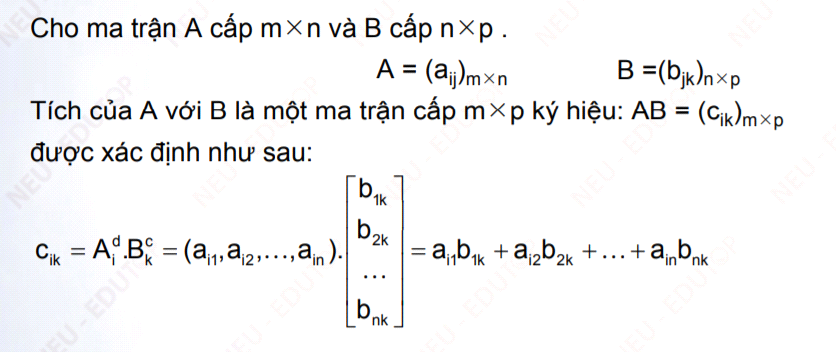


Figure 17 Công thức nhân ma trận

Phương thức chuyển vị ma trận.

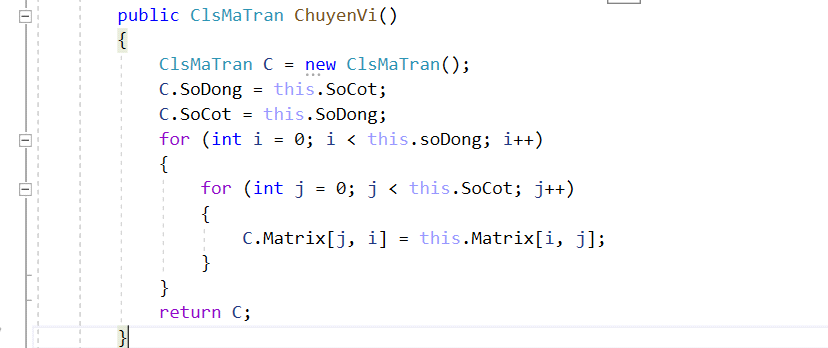


Figure 18 Chuyển vị ma trận

Ma trận chuyển vị được tính bằng cách hoán đổi vị trí thứ I và j lại với nhau.

Số dòng và số cột tương ứng của ma trận chuyển vị sẽ bằng số cột và số dòng của ma trận nguồn.

Phương thức tính độ dài từng vector trong ma trận.

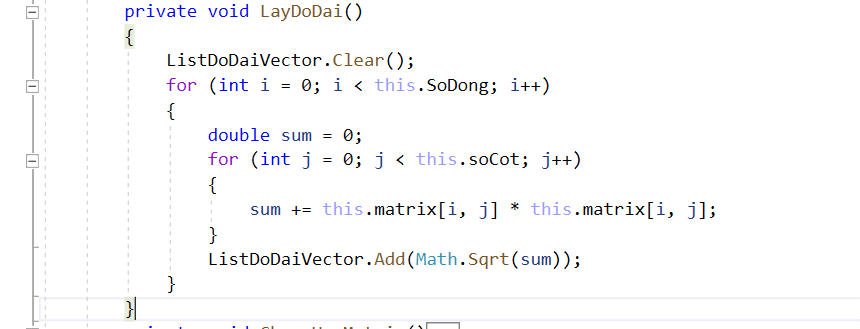


Figure 19 Lấy độ dài

Độ dài của vector được tính theo cách sau:

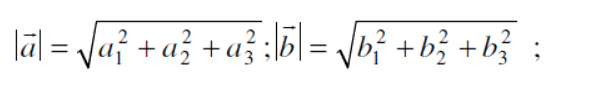


Figure 20 Công thức tính độ dài

Tương tự như các vector trong ma trận. Trước tiên, ta tính tổng bình phương các phần tử trong vector, sau đó lấy căn bậc hai của tổng đó.

Độ dài của vector sẽ được lưu trong list ListDoDaiVector.

Phương thức chuẩn hóa ma trận.

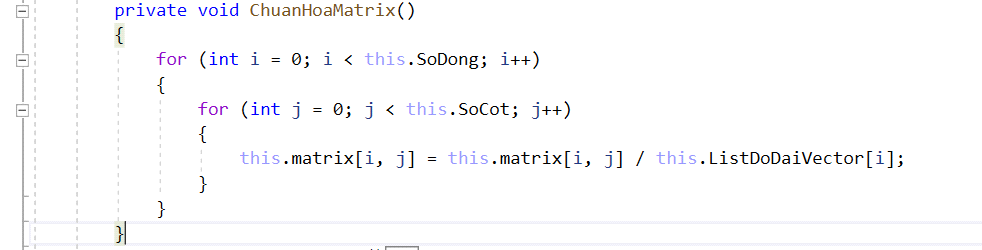


Figure 21 Chuẩn hóa ma trận

Chuẩn hóa ma trận tính bằng cách lấy từng giá trị của ma trận chia cho độ dài vector của hàng đó.

Phương thức tính độ tương đồng cosin giữa các vector trong ma trận.

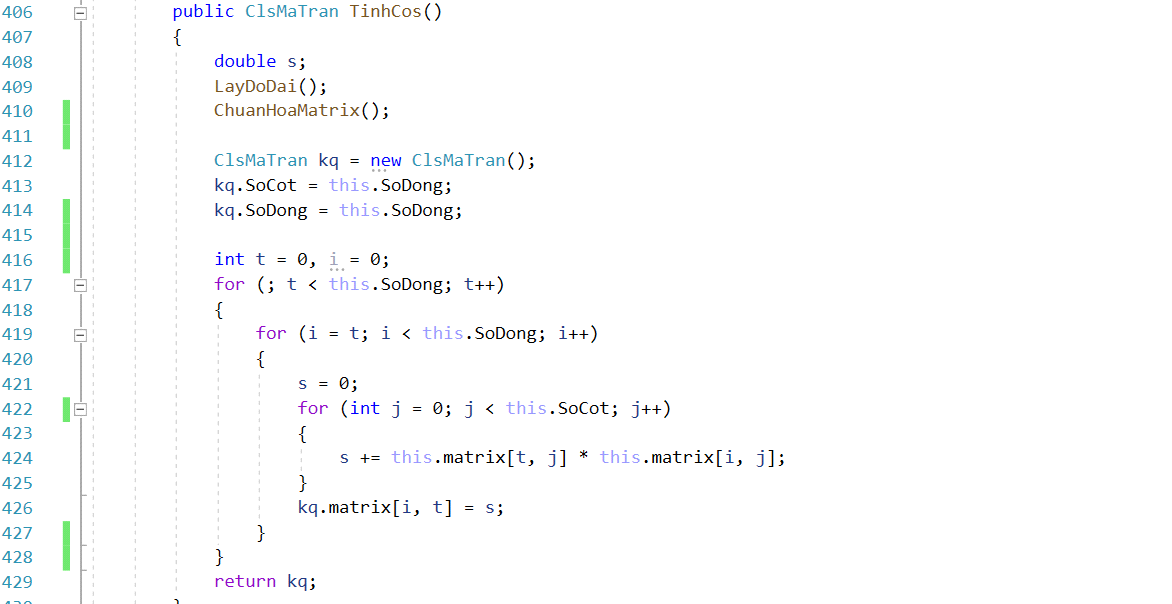


Figure 22 Độ tương đồng cosin giữa các vecto trogn ma trận

Đầu tiên ta cần tìm độ dài của từng vector trong ma trận.

Sau đó ta sẽ chuẩn hóa ma trận.

Tiếp đến ta sẽ tính tổng của tích từng phần tử giữa hai vector với nhau.

Độ tương đồng giữa hai vector sẽ được lưu trong ma trận kết quả, bằng với vị trí dòng i, cột j tương ứng.

Ví dụ: cho ma trận:

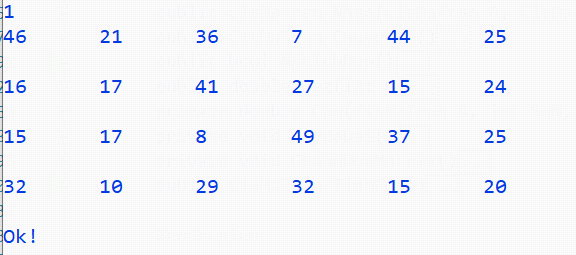


Figure 23 Ví dụ tính độ tương đối

Ma trận sau khi được chuẩn hóa:

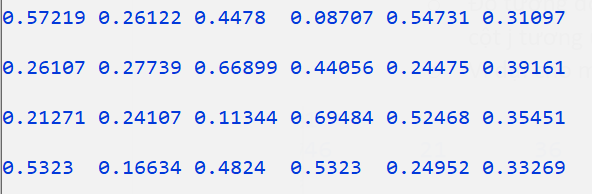


Figure 24 Ví dụ tính độ tương đối

Ma trận lưu kết quả tính độ tương đồng cosin:

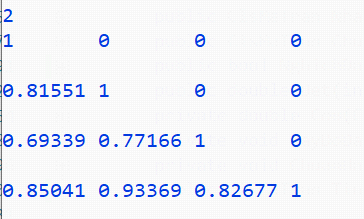


Figure 25 Ví dụ tính độ tương đối

Phương thức nghịch đảo ma trận.

Để tìm ra ma trận nghịch đảo thì ta cần tìm ma trận phù hợp và tính định thức.

Rồi sau đó lấy ma trận phù hợp chia cho định thức là ra ma trận nghịch đảo.

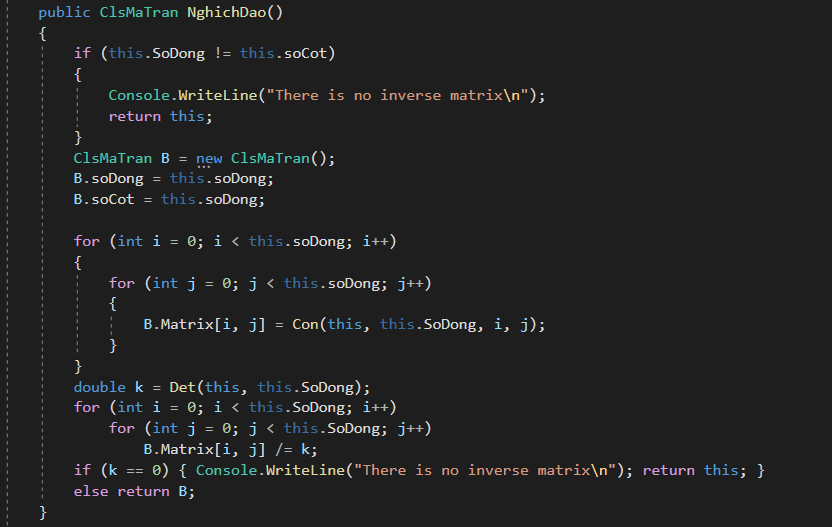


Figure 26 Ma trận nghịch đảo

Tìm ma trận phù hợp và tính định thức.

Tìm ma trận phù hợp, ứng với mỗi phần tử trong ma trận ta sẽ bỏ đi dòng với cột tương ứng trong ma trận đó. Rồi đi tính định thức của ma trận còn lại.

VD:

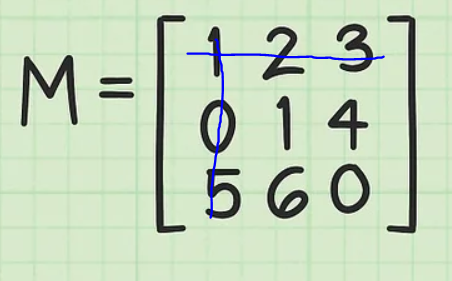


Figure 27 Ví dụ nghịch đảo

Với M[0,0] thì ta bỏ đi dòng 0 cột 0, rồi sau đó ta đi tính định thức cho ma trận với kích thước n-1( ứng với trong minh họa là n=3-1=2). Nếu tổng hai vị trí là số chẵn thì sẽ trả về định thức dương và ngược lại.

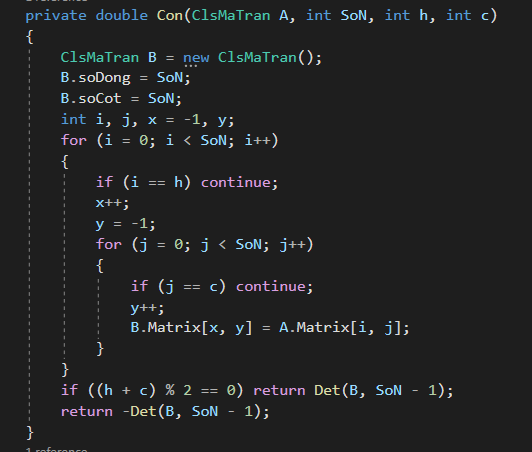


Figure 28 Tìm ma trận con

Tương ứng ta được dòng code trên. Với x,y là những vị trí cần lấy của ma trận A để đưa vào ma trận B để đi tính định thức. Và nếu chẵn thì ta sẽ trả về 1 định thức dương ngược lại.

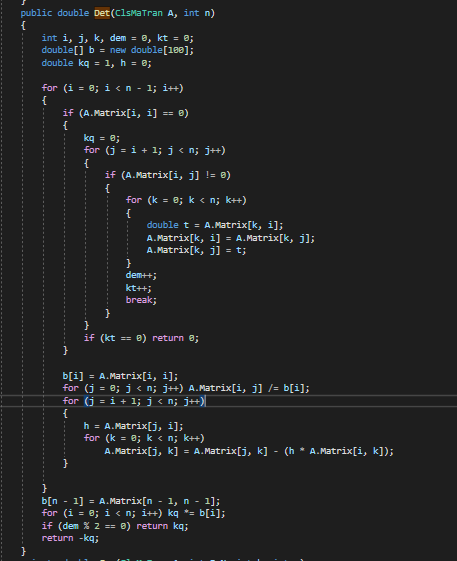


Figure 29 Tính định thức

Để tính định thức cần truyền vào một ma trận( ma trận con tìm được ở trên)

Trong lúc tính sẽ đi kiểm tra đường chéo chính xem có giá trị nào bằng không không. Nếu có sẽ swap cho phần tử cùng hàng mà khác không.

2.6 Sản phẩm của đồ án

Một ứng dụng tính toán ma trận.

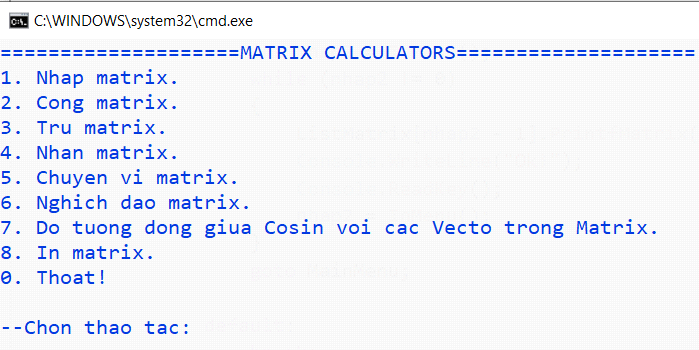


Figure 30 Các chức năng

CHƯƠNG 3 : KẾT LUẬN

3.1 Kết luận

Với mục tiêu đã đặt ra thì nhóm tự tin đã hoàn thành được hơn 90% kế hoạch ban đầu. Nhóm vận dụng kiến thức đã học về ma trận để mô phỏng lại đầy đủ các chức năng cơ bản của ma trận.

3.2 Ưu điểm

Ứng dụng đơn giản, dễ sử dụng, khả năng tương thích cao, không yêu cầu cấu hình cao để chạy.

Vì ứng dụng sử dụng tính toán ma trận có kích thước rất lớn, nên nhóm quyết định xây dựng ứng dụng chạy trên giao diện console để giúp người dùng thao tác đơn giản hơn, việc nhập có thể thực hiện trên bàn phím nên giao diện console là một lựa chọn tốt.

3.3 Nhược điểm

Vì ứng dụng chạy trên giao diện console, nên việc thao tác cần độ chính xác cao. Giao diện chưa bắt mắt, chỉ phù hợp cho một số người dùng.

Đối với ma trận có kích thước rất lớn, số lượng dòng và cột có thể vượt quá số dòng và cột của Excel, nên việc lưu ma trận vào file có thể bị sai sót.

3.4 Đề xuất phương án cải thiện

Có thể xây dựng ứng dụng trên các nền tảng như Winform và WPF để giao diện người dùng trở nên gần gũi hơn.

Đối với ma trận có kích thước rất lớn chúng ta nên lưu vào file Txt, vì các phần tử của ma trận sẽ được lưu trên mỗi dòng riêng biệt nên sẽ không bị giới hạn số phần tử trên một hàng.

PHỤ LỤC

Bảng 1: Bảng phân công

TÀI LIỆU THAM KHẢO

[1] - https://vi.wikipedia.org/wiki/C%E1%BB%9D\_ca-r%C3%B4

[2] -https://vi.wikipedia.org/wiki/Ng%C4%83n\_x%E1%BA%BFp

[3] - https://www.howkteam.vn/course/lap-trinh-game-caro-voi-c-winform-14