TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**HUỲNH VĂN DŨNG - 52300190**

**CAO TUẤN CƯỜNG - 52300183**

**TIỂU LUẬN GIỮA KỲ**

**ĐẠI SỐ TUYẾN TÍNH CHO  
CÔNG NGHỆ THÔNG TIN**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**HUỲNH VĂN DŨNG - 52300190**

**CAO TUẤN CƯỜNG - 52300183**

**TIỂU LUẬN GIỮA KỲ**

**ĐẠI SỐ TUYẾN TÍNH CHO**

**CÔNG NGHỆ THÔNG TIN**

Người hướng dẫn

**ThS. Phan Hoàng Phúc Vinh**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

**LỜI CẢM ƠN**

Lời đầu tiên, em xin trân trọng cảm ơn giảng viên **ThS. Phan Hoàng Phúc Vinh** người đã trực tiếp chỉ bảo, hướng dẫn em trong quá trình hoàn thành bài tiểu luận này.

Em cũng xin được gửi lời cảm ơn đến quý thầy, cô giáo trường Đại Học Tôn Đức Thắng, đặc biệt là các thầy, cô khoa công nghệ thông tin - những người đã truyền lửa và giảng dạy kiến thức cho em suốt thời gian qua.

Mặc dù đã có những đầu tư nhất định trong quá trình làm bài song cũng khó có thể tránh khỏi những sai sót, em kính mong nhận được ý kiến đóng góp của quý thầy cô để bài tiểu luận được hoàn thiện hơn.

Em xin chân thành cảm ơn!

*TP. Hồ Chí Minh, ngày 22 tháng 03 năm 2024*

*Tác giả Tác giả*

*Ảnh có chứa chữ viết tay, thư pháp, màu đen, đen và trắng

Mô tả được tạo tự động Ảnh có chứa chữ viết tay, thư pháp

Mô tả được tạo tự động*

*Cao Tuấn Cường Huỳnh Văn Dũng*

**TIỂU LUẬN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là Tiểu luận của riêng chúng tôi và được sự hướng dẫn khoa học của ThS. Phan Hoàng Phúc Vinh   Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong Tiểu luận còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung Tiểu luận của mình**. Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 22 tháng 03 năm 2024*

*Tác giả Tác giả*

*Ảnh có chứa chữ viết tay, thư pháp, màu đen, đen và trắng

Mô tả được tạo tự động Ảnh có chứa chữ viết tay, thư pháp

Mô tả được tạo tự động*

*Cao Tuấn Cường Huỳnh Văn Dũng*

# **TÓM TẮT**

Các kiến thức toán học đã được học và ngôn ngữ lập trình Python chính là những đối tượng và phạm vi nghiên cứu chính để chúng em có thể hoàn thành bài tiểu luận này. Chúng em cũng đã vận một số phương pháp nghiên cứu như phương pháp logic, phương pháp thực nghiệm, phương pháp tổng kết, phương pháp nghiên cứu định lượng. Đó là những cách tiếp cận vấn đề mà chúng em cảm thấy phù hợp nhất cho vấn đề được đặt ra

Tổng kết lại, qua bài tiểu luận này, chúng em và người đọc đều có thể học được thêm nhiều bài học cho bản thân: tư duy giải quyết vấn đề, các kiến thức toán học cơ bản, khả năng lập trình, khả năng sắp xếp và liên kết,...

Chương 1: giới thiệu ngôn ngữ Python, môi trường Pycharm và các hàm được sử dụng trong bài, nêu ra cách giải quyết vấn đề và các - thực hiện.

Chương 2: trình bày mã nguồn Python giải quyết vấn đề và kết quả cuối cùng của các câu hỏi.

# **DANH SÁCH HÌNH ẢNH**

[Hình 1. 1 Kết quả các ma trận. 12](#_Toc164411332)

[Hình 1. 2 Kết quả câu a. 13](#_Toc164411333)

[Hình 1. 3 Kết quả câu b. 14](#_Toc164411334)

[Hình 1. 4 Kết quả câu c. 15](#_Toc164411335)

[Hình 1. 5 Kết quả câu d. 15](#_Toc164411336)

[Hình 1. 6 Kết quả câu e. 15](#_Toc164411337)

[Hình 1. 7 Kết quả câu f. 16](#_Toc164411338)

[Hình 1. 8 Kết quả câu g. 18](#_Toc164411339)

[Hình 1. 9 Kết quả câu h. 20](#_Toc164411340)

# **MỤC LỤC**

[**TÓM TẮT** iii](#_Toc164411374)

[**DANH SÁCH HÌNH ẢNH** iv](#_Toc164411375)

[**MỤC LỤC** v](#_Toc164411376)

[**CHƯƠNG I: CƠ SỞ LÝ THUYẾT** 1](#_Toc164411377)

[1.1 Giới thiệu 1](#_Toc164411378)

[1.1.1 Ngôn ngữ 1](#_Toc164411379)

[1.1.2 Môi trường 1](#_Toc164411380)

[1.1.3 Thư viện 2](#_Toc164411381)

[1.2 Phương pháp giải bài tập 3](#_Toc164411382)

[1.2.1 Câu trả lời chung cho câu 1 từ a đến h 3](#_Toc164411383)

[1.2.2 Câu d: In vector chứa các số nguyên lẻ trong ma trận A. 3](#_Toc164411384)

[1.2.3 Câu e: In vector chứa các số nguyên tố trong ma trận A. 3](#_Toc164411385)

[1.2.4 Câu f: In ma trận D=CB chứa các phần tử nghịch đảo của ma trận D. 4](#_Toc164411386)

[1.2.5 Câu g: In các hàng có số lượng số nguyên tố lớn nhất. 4](#_Toc164411387)

[1.2.6 Câu h: In các hàng có chuỗi số lẻ liên tục dài nhất. 4](#_Toc164411388)

[**CHƯƠNG II: MÃ NGUỒN VÀ KẾT QUẢ ĐẦU RA** 5](#_Toc164411389)

[2.1 Mã nguồn 5](#_Toc164411390)

[2.1.1 Câu d: In vector chứa các số nguyên lẻ trong ma trận A. 7](#_Toc164411391)

[2.1.2 Câu e: In vector chứa các số nguyên tố trong ma trận A. 8](#_Toc164411392)

[2.1.3 Câu f: In ma trận D=CB chứa các phần tử nghịch đảo của ma trận D. 9](#_Toc164411393)

[2.1.4 Câu g: In các hàng có số lượng số nguyên tố lớn nhất. 10](#_Toc164411394)

[2.1.5 Câu h: In các hàng có chuỗi số lẻ liên tục dài nhất. 11](#_Toc164411395)

[2.2 Kết quả đầu ra 12](#_Toc164411396)

[2.2.1 Câu a: Tính *A+* 13](#_Toc164411397)

[2.2.2 Câu b: Tính  14](#_Toc164411398)

[2.2.3 Câu c: In ma trận mới có các hàng lẻ của ma trận A. 15](#_Toc164411399)

[2.2.4 Câu d: In vector chứa các số nguyên lẻ trong ma trận A. 15](#_Toc164411400)

[2.2.5 Câu e: In vector chứa các số nguyên tố trong ma trận A. 15](#_Toc164411401)

[2.2.6 Câu f: In ma trận D=CB chứa các phần tử nghịch đảo của ma trận D. 16](#_Toc164411402)

[2.2.7 Câu g: In các hàng có số lượng số nguyên tố lớn nhất. 17](#_Toc164411403)

[2.2.8 Câu h: In các hàng có chuỗi số lẻ liên tục dài nhất. 19](#_Toc164411404)

[**TÀI LIỆU THAM KHẢO** 21](#_Toc164411405)

# **CHƯƠNG I: CƠ SỞ LÝ THUYẾT**

## 1.1 Giới thiệu

### 1.1.1 Ngôn ngữ

***Python*** là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình và là ngôn ngữ lập trình dễ học; được dùng rộng rãi trong phát triển trí tuệ nhân tạo. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu.

### 1.1.2 Môi trường

***PyCharm*** là môi trường phát triển tích hợp đa nền tảng (IDE) được phát triển bởi Jet Brains và được thiết kế đặc biệt cho Python. PyCharm có mặt trên cả 3 nền tảng Windows, Linux và Mac OS. Pycharm là một giải pháp phù hợp cho Python developers vì IDE này hỗ trợ nhiều extensions, môi trường ảo (Virtual Environment), nhiều tính năng thông minh như bộ code completion, tự động thụt lề, phát hiện văn bản trùng lặp và kiểm tra lỗi. Ngoài ra còn có các tính năng tìm kiếm mã nguồn thông minh để tìm kiếm từng từ một trong nháy mắt.

**Visual Studio Code** , còn thường được gọi là **VS Code**,  là [trình soạn thảo mã nguồn](https://en.wikipedia.org/wiki/Source-code_editor) được phát triển bởi [Microsoft](https://en.wikipedia.org/wiki/Microsoft) cho [Windows](https://en.wikipedia.org/wiki/Windows), [Linux](https://en.wikipedia.org/wiki/Linux) , [macOS](https://en.wikipedia.org/wiki/MacOS) và [trình duyệt web](https://en.wikipedia.org/wiki/Web_browser).  Các tính năng bao gồm hỗ trợ [gỡ lỗi](https://en.wikipedia.org/wiki/Debugging) , [tô sáng cú pháp](https://en.wikipedia.org/wiki/Syntax_highlighting) , [hoàn thành mã thông minh](https://en.wikipedia.org/wiki/Intelligent_code_completion) , [đoạn mã](https://en.wikipedia.org/wiki/Snippet_(programming)) , [tái cấu trúc mã](https://en.wikipedia.org/wiki/Code_refactoring) và [kiểm soát phiên bản](https://en.wikipedia.org/wiki/Version_control) nhúng bằng [Git](https://en.wikipedia.org/wiki/Git) . Người dùng có thể thay đổi [chủ đề](https://en.wikipedia.org/wiki/Theme_(computing)) , [phím tắt](https://en.wikipedia.org/wiki/Keyboard_shortcut) , tùy chọn và cài đặt [các tiện ích mở rộng](https://en.wikipedia.org/wiki/Plug-in_(computing)) bổ sung chức năng.

### 1.1.3 Thư viện

***SymPy*** là một thư viện toán học biểu tượng Python. Nó mong muốn trở thành một hệ thống đại số máy tính (CAS) đầy đủ tính năng trong khi vẫn giữ mã cơ bản nhất có thể để dễ hiểu và có thể mở rộng dễ dàng. Nó được viết hoàn toàn bằng Python. Nó đơn giản để sử dụng vì nó chỉ dựa vào mpmath, một thư viện Python thuần túy cho số học dấu phẩy động tùy ý. Là một thư viện, nó được tạo ra với sự chú trọng đáng kể đến tính khả dụng trong tâm trí. Khả năng mở rộng là rất quan trọng trong việc thiết kế giao diện chương trình ứng dụng (API) của nó.

***Numpy*** là một thư viện trong Python mã nguồn mở chủ yếu được sử dụng để thao tác và xử lý dữ liệu dưới dạng mảng. Vì vậy, nó khá dễ học và hoạt động nhanh, hoạt động tốt với các thư viện khác, có nhiều chức năng tích hợp và cho phép bạn làm các phép toán ma trận.

***Math*** được định nghĩa trong toán học là một hàm toán phổ biến nhất bao gồm các hàm lượng giác, hàm số, hàm logarit,..Ngoài ra nó cũng định nghĩa hai hằng số toán học tức là số Pie và Euler,.v.v.

***Matplotlib*** là một [thư viện](https://en.wikipedia.org/wiki/Library_(computer_science))[vẽ đồ thị](https://en.wikipedia.org/wiki/Plotter) cho [ngôn ngữ lập trình](https://en.wikipedia.org/wiki/Programming_language)[Python](https://en.wikipedia.org/wiki/Python_(programming_language)) và phần mở rộng [toán học số](https://en.wikipedia.org/wiki/Numerical_analysis)[NumPy](https://en.wikipedia.org/wiki/NumPy) của nó . Nó cung cấp [API](https://en.wikipedia.org/wiki/API)[hướng đối tượng để nhúng các biểu đồ vào các ứng dụng bằng cách sử dụng](https://en.wikipedia.org/wiki/Object-oriented_programming)[các bộ công cụ GUI](https://en.wikipedia.org/wiki/GUI_toolkit) có mục đích chung như [Tkinter](https://en.wikipedia.org/wiki/Tkinter) , [wxPython](https://en.wikipedia.org/wiki/WxPython) , [Qt](https://en.wikipedia.org/wiki/Qt_(software)) hoặc [GTK](https://en.wikipedia.org/wiki/GTK) . Ngoài ra còn có giao diện "pylab" [thủ tục](https://en.wikipedia.org/wiki/Procedural_programming) dựa trên [máy trạng thái](https://en.wikipedia.org/wiki/State_machine) (như [OpenGL](https://en.wikipedia.org/wiki/OpenGL) ), được thiết kế gần giống với giao diện của [MATLAB](https://en.wikipedia.org/wiki/MATLAB) , mặc dù việc sử dụng nó không được khuyến khích. [[3]](https://en.wikipedia.org/wiki/Matplotlib#cite_note-3)[SciPy](https://en.wikipedia.org/wiki/SciPy) sử dụng Matplotlib.

## 1.2 Phương pháp giải bài tập

### 1.2.1 Câu trả lời chung cho câu 1 từ a đến h

Tạo một ma trận A 10x10 với phần tử ngẫu nhiên trong khoảng [1,100], một ma trận B 2x10 với số nguyên ngẫu nhiên với phần tử trong khoảng [1,20] và một ma trận C 10x2 với phần tử trong khoảng [1,20]

- Sử dụng hàm [numpy.random.randin](https://numpy.org/doc/stable/reference/random/generated/numpy.random.randint.html)t để tạo các ma trận ngẫu nhiên  
- Sử dụng câu lệnh print để hiển thị các ma trận đã tạo

### 1.2.2 Câu d: In vector chứa các số nguyên lẻ trong ma trận A.

- Hàm [***flatten***](https://numpy.org/doc/stable/reference/generated/numpy.ndarray.flatten.html)***()*** biến mảng đa chiều thành mảng một chiều,  
- Loại bỏ các số nguyễn chẵn trong bài bằng cách sử dụng toán tử (%2!=0)

- Hàm ***print()*** để in vector chứa các số nguyên lẻ trong ma trận A.

### 1.2.3 Câu e: In vector chứa các số nguyên tố trong ma trận A.

- Hàm ***np.aray*** để biến danh sách thành vector

- Hàm kiểm tra 1 số có phải là [***số nguyên tố***](https://vi.wikipedia.org/wiki/Số_nguyên_tố) hay không nếu là [***số nguyên tố***](https://vi.wikipedia.org/wiki/Số_nguyên_tố) thì thêm vào vecto rỗng đã tạo sẵn

- Hàm ***print()*** để in vector chứa các số nguyên tố trong ma trận A.

### 1.2.4 Câu f: In ma trận D=CB chứa các phần tử nghịch đảo của ma trận D.

- Sử dụng thư viện [***np.dot***](https://numpy.org/doc/stable/reference/generated/numpy.dot.html)***,*** [***np.array***](https://numpy.org/doc/stable/reference/generated/numpy.array.html) để tạo ma trận D theo yêu cầu

- Hàm sử dụng vòng lặp for để kiểm tra phải hàng lẻ hay không, nếu là hàng lẻ thì đảo ngược các giá trị trong ma trận.

- Hàm ***print()*** để in ra ma trận D

### 1.2.5 Câu g: In các hàng có số lượng số nguyên tố lớn nhất.

- Tạo một biến đếm để lưu vào danh sách ấy những hàng cần in ra.

- Tạo thêm một biến khác đếm lượng số nguyên tố trong một dòng.

- Sử dụng vòng lặp for để xử lý từng hàng của ma trận ấy, và sử dụng câu lệnh if để xử lý biến đếm số lượng để tìm ra dòng có lượng số nguyên tố lớn nhất .

- Sử dụng vòng lặp for để thêm vào danh sách các hàng có lượng số nguyên tố lớn nhất nếu hàng có số lượng số nguyên tố bằng biến số lượng hiện tại.

- Hàm print() để in các hang có số lượng số nguyên tố lớn nhất.

### 1.2.6 Câu h: In các hàng có chuỗi số lẻ liên tục dài nhất.

- Tạo một biến đếm độ dài của chuỗi số lẻ, và một biến max để lưu lại chuỗi có lượng số lẻ lớn nhất

- Sử dụng vòng lặp for để xử lý từng hàng trong ma trận khi chuỗi số lẻ của không còn tiếp tục thì so sánh với chuỗi dài nhất cho đến hiện tại.

- Khi tìm ra hàng có chuỗi số lẻ dài nhất, tìm các hàng có chuỗi số lẻ cùng có độ dài với chuỗi lớn nhất hiện tại, thêm các các hàng đấy vào danh sách rỗng.

- Hàm ***print()*** in các hàng có chuỗi số lẻ liên tục dài nhất.

# **CHƯƠNG II: MÃ NGUỒN VÀ KẾT QUẢ ĐẦU RA**

## 2.1 Mã nguồn

|  |
| --- |
| import numpy as np  prime = []  #tạo ma trận A  A = np.random.randint(1, 101, size=(10, 10))  #tạo ma trận B  B = np.random.randint(1, 21, size=(2, 10))  #tạo ma trận C  C = np.random.randint(1, 21, size=(10, 2))  #in ra các ma trận ban đầu  print("Ma tran A (10x10): ")  print(A)  print("\nMa tran B (2x10): ")  print(B)  print("\nMa tran C (10x2): ")  print(C)  #câu 1a  #sử dụng hàm hoán vị các ma trận transpose(.T) để thực hiện phép tính  A\_transpose = A.T  B\_transpose = B.T  C\_transpose = C.T  result = A + A\_transpose + np.dot(C, B) + np.dot(B\_transpose, C\_transpose)  print("Ket qua cau 1a:")  print(result)  #cau 1b  result1 = 0  for i in range(0,10):      result1 += (A / (i + 10)) \*\* (i + 1)  print("Ket qua cau 1b:")  print(result1)  #cau 1c  print("In ra cac hang le cua ma tran A:")  print(A[::2])# sử dụng A[::2] để bỏ các hàng chẵn đi |

### 2.1.1 Câu d: In vector chứa các số nguyên lẻ trong ma trận A.

|  |
| --- |
| #cau 1d  #flatten là hàm dùng để biến mảng đa chiều thành mảng một chiều, ta dùng hàm này để lọc và in ra các số lẻ trong ma trận  print("vector cac so le cua ma tran A:")  print(A.flatten()[A.flatten() %2 !=0]) |

### 2.1.2 Câu e: In vector chứa các số nguyên tố trong ma trận A.

|  |
| --- |
| #Câu 1e  A\_vector = A.flatten()  prime\_array = []  #hàm kiểm tra có phải số nguyên tố không  def is\_prime(num):      if num <= 1:          return False      elif num <= 3:          return True      elif num % 2 == 0 or num % 3 == 0:          return False      i = 5      while i \* i <= num:          if num % i == 0 or num % (i + 2) == 0:              return False          i += 6      return True  def prime():   for num in A\_vector:       if is\_prime(num): #sử duụng hàm để xét có phải só nguyên tố không           prime\_array.append(num) #nếu là số nguyên tố thì thêm vào danh sách   print(prime\_array)  print("Ma tran cac so nguyen to : ")  print(prime()) |

### 2.1.3 Câu f: In ma trận D=CB chứa các phần tử nghịch đảo của ma trận D.

|  |
| --- |
| #câu 1f  D = np.dot(np.array(C), np.array(B)) #ma trận D theo đề bài  def Dmatrix():   for i in range(len(D)):      if i % 2 != 0:  #kiểm tra có phải hàng lẻ không          D[i] = D[i][::-1]  # đảo ngược các giá trị trong ma trận   print(D)  print("Ma tran D:")  print(Dmatrix()) |

### 2.1.4 Câu g: In các hàng có số lượng số nguyên tố lớn nhất.

|  |
| --- |
| #câu 1g  max\_prime\_count = 0  max\_prime\_rows = []  #ta cho một biến đếm để xác định những dòng có lượng số nguyên tố lớn nhất rôi thêm vào một danh sách  for i, row in enumerate(A):      prime\_count = sum(1 for num in row if is\_prime(num)) #đếm lượng số nguyên tố trong một dòng      if prime\_count > max\_prime\_count: #nếu lượng  số nguyên tố của dòng đó lớn hơn hàng lớn nhất hiện tại thì nó sẽ thay thế để trở thành hàng lớn nhất          max\_prime\_count = prime\_count          max\_prime\_rows = [i]      elif prime\_count == max\_prime\_count:  #thêm vào danh sách các hàng có lượng số nguyên tố lớn nhất nếu hàng có số lượng số nguyên tố bằng biến max hiện tại          max\_prime\_rows.append(i)  #in ra các hàng có nhiều số nguyên tố nhất  print("Hang co nhieu so Nguyen to nhat la")  for i in max\_prime\_rows:      print(A[i]) |

### 2.1.5 Câu h: In các hàng có chuỗi số lẻ liên tục dài nhất.

|  |
| --- |
| #cau 1h  #hàm tìm dòng có chuỗi số lẻ dài nhất  def longest\_contiguous\_odd\_sequence(row):      max\_length = 0      current\_length = 0      for num in row:          if num % 2 != 0:              current\_length += 1 #cho một biến đếm độ dài của chuỗi số lẻ, và một biến max để lưu lại chuỗi có lượng số lẻ lớn nhất          else:              max\_length = max(max\_length, current\_length) #khi chuỗi không còn tiếp tục thì so sánh với chuỗi dài nhất cho đến hiện tại              current\_length = 0      max\_length = max(max\_length, current\_length)      return max\_length  max\_sequence\_length = 0  max\_sequence\_rows = []  for i, row in enumerate(A):      sequence\_length = longest\_contiguous\_odd\_sequence(row) #sử dụng hàm tìm cdòng có chuỗi số lẻ có dộ dài lớn nhất ở trên      if sequence\_length > max\_sequence\_length:  #tiếp tục tìm các chuôi có chuỗi số lẻ cùng có độ dài với chuỗi lớn nhất hiện tại          max\_sequence\_length = sequence\_length #thêm các chuỗi đấy vào danh sách để in ra          max\_sequence\_rows = [i]      elif sequence\_length == max\_sequence\_length:          max\_sequence\_rows.append(i)  print("Hang co chuoi so le lien ke lon nhat cua ma tran A la:")  for i in max\_sequence\_rows:      print(A[i]) |

## 2.2 Kết quả đầu ra

|  |
| --- |
|  |

Hình 1. 1 Kết quả các ma trận.

### 2.2.1 Câu a: Tính *A+*

|  |
| --- |
|  |

Hình 1. 2 Kết quả câu a.

### 2.2.2 Câu b: Tính

|  |
| --- |
|  |

Hình 1. 3 Kết quả câu b.

### 2.2.3 Câu c: In ma trận mới có các hàng lẻ của ma trận A.

|  |  |
| --- | --- |
|  |  |

Hình 1. 4 Kết quả câu c.

### 2.2.4 Câu d: In vector chứa các số nguyên lẻ trong ma trận A.

|  |  |
| --- | --- |
|  |  |

Hình 1. 5 Kết quả câu d.

### 2.2.5 Câu e: In vector chứa các số nguyên tố trong ma trận A.

|  |  |
| --- | --- |
|  |  |

Hình 1. 6 Kết quả câu e.

### 2.2.6 Câu f: In ma trận D=CB chứa các phần tử nghịch đảo của ma trận D.

|  |
| --- |
|  |

Hình 1. 7 Kết quả câu f.

### 2.2.7 Câu g: In các hàng có số lượng số nguyên tố lớn nhất.

Lần 1:

|  |  |
| --- | --- |
| Ma trận A(10x10)  [76 9 22 35 69 85 9 90 82 94]  [46 29 100 68 85 25 98 91 40 70]  [36 78 1 67 75 24 66 22 70 66]  [75 90 62 82 33 26 55 47 72 22]  [22 19 54 42 46 57 38 14 56 83]  [90 56 91 32 84 21 21 43 16 54]  [68 59 82 91 35 56 60 74 65 65]  [31 95 51 87 3 29 69 9 78 26]  [66 13 82 58 97 94 76 94 24 79]  [70 96 74 4 95 87 82 79 35 37] | # dòng 1 không có số nguyên tố  # dòng 2 có 1 số nguyên tố  # dòng 3 có 1 số nguyên tố  # dòng 4 không có số nguyên tố  # dòng 5 có 2 số nguyên tố  # dòng 6 có 1 số nguyên tố  # dòng 7 không có số nguyên tố  # dòng 8 có 3 số nguyên tố => nhiều nhất  # dòng 9 có 3 số nguyên tố => nhiều nhất  # dòng 10 có 2 số nguyên tố |
|  |  |

Lần 2:

|  |  |
| --- | --- |
| Ma trận A(10x10) lần 2: [79 31 87 65 67 2 75 81 33 33]  [100 60 1 79 35 64 56 69 33 36]  [81 75 82 5 49 8 27 2 79 41]  [30 25 8 68 32 48 61 89 54 41]  [26 3 38 71 99 97 29 94 51 8 ]  [96 68 9 22 90 80 75 73 71 62]  [72 98 50 37 44 41 71 2 35 4 ]  [41 86 32 99 64 96 60 20 49 72]  [89 11 26 84 65 36 61 84 11 43]  [63 95 8 76 51 41 94 76 89 79] | # dòng 1 có 3 số nguyên tố  # dòng 2 có 1 số nguyên tố  # dòng 3 có 4 số nguyên tố  # dòng 4 có 3 số nguyên tố  # dòng 5 có 4 số nguyên tố  # dòng 6 có 2 số nguyên tố  # dòng 7 có 2 số nguyên tố  # dòng 8 có 1 số nguyên tố  # dòng 9 có 5 số nguyên tố => nhiều nhất  # dòng 10 có 3 số nguyên tố |
|  |  |

|  |
| --- |
|  |

Hình 1. 8 Kết quả câu g.

### 2.2.8 Câu h: In các hàng có chuỗi số lẻ liên tục dài nhất.

Lần 1:

|  |  |
| --- | --- |
| Ma trận A(10x10)  [76 9 22 35 69 85 9 90 82 94]  [46 29 100 68 85 25 98 91 40 70]  [36 78 1 67 75 24 66 22 70 66]  [75 90 62 82 33 26 55 47 72 22]  [22 19 54 42 46 57 38 14 56 83]  [90 56 91 32 84 21 21 43 16 54]  [68 59 82 91 35 56 60 74 65 65]  [31 95 51 87 3 29 69 9 78 26]  [66 13 82 58 97 94 76 94 24 79]  [70 96 74 4 95 87 82 79 35 37] | # dòng 1 có 4 số lẻ liền kề  # dòng 2 có 3 số lẻ, 2 số lẻ liền kề  # dòng 3 có 3 số lẻ liền kề  # dòng 4 có 4 số lẻ, 2 số lẻ liền kề  # dòng 5 có 3 số lẻ không liền kề  # dòng 6 có 4 số lẻ, 3 số lẻ liền kề  # dòng 7 có 5 số lẻ, 2 số lẻ liền kề  # dòng 8 có 8 số lẻ liền kề => nhiều nhất  # dòng 9 có 3 số lẻ không liền kề  # dòng 10 có 4 số lẻ, 3 số lẻ liền kề |
|  |  |

|  |  |
| --- | --- |
|  |  |

Lần 2:

|  |  |
| --- | --- |
| Ma trận A(10x10) lần 2: [88 24 88 5 18 21 9 99 2 35]  [7 100 40 17 32 69 46 49 100 90]  [93 41 29 28 86 52 70 35 16 17]  [95 48 90 23 16 27 84 7 51 34]  [93 37 74 66 50 19 94 22 63 46]  [43 26 5 66 23 21 96 14 36 79]  [25 10 39 98 38 59 22 99 7 16]  [31 32 77 78 78 22 84 67 22 67]  [31 27 92 71 33 33 50 36 57 80]  [78 41 65 52 65 55 46 70 27 68 ] | # dòng 1 có 5 số lẻ, 3 số lẻ liền kề  # dòng 4 có 5 số lẻ, 0 số lẻ liền kề  # dòng 3 có 5 số lẻ, 3 số lẻ liền kề  # dòng 4 có 4 số lẻ, 2 số lẻ liền kề  # dòng 5 có 4 số lẻ, 3 số lẻ liền kề  # dòng 6 có 5 số lẻ, 2 số lẻ liền kề  # dòng 7 có 5 số lẻ, 0 số lẻ liền kề  # dòng 8 có 4 số lẻ, 0 số lẻ liền kề  # dòng 9 có 6 số lẻ, 3 số lẻ liền kề  # dòng 10 có 4 số lẻ, 3 số lẻ liền kề |
|  |  |

|  |
| --- |
|  |

Hình 1. 9 Kết quả câu h.

# **TÀI LIỆU THAM KHẢO**

**Tiếng Việt**

[[1]](https://giaotrinhpdf.com/giao-trinh-dai-so-tuyen-tinh.html#gsc.tab=0) Ngô Việt Trung, [2023], Giáo trình đại số tuyến tính, Hà Nội

[[2]](https://bktoanthu.com/tailieu/181/Ch%C6%B0%C6%A1ng%203:%20Kh%C3%B4ng%20gian%20vecto%20(BK%C4%90CMP%20bi%C3%AAn%20so%E1%BA%A1n)) Phạm Thanh Tùng, Đại số tuyến tính, Hà Nội

**Tiếng Anh**

[[1]](https://pdfexist.com/download/3812534-matrix-algebra-for-linear-models.pdf) Marvin H.J.Gruber, [2013], Matrix Algebra for Linear Models, USA

[[2]](https://archive.org/details/matricesvectorsp0000brow)  Brown, William C. (1991), [*Matrices and vector spaces*](https://archive.org/details/matricesvectorsp0000brow), New York