## **ĐẠI HỌC QUỐC GIA TP.HCM**

## **ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

## KHOA CÔNG NGHỆ THÔNG TIN





## BÁO CÁO ĐỒ ÁN

Môn: Cấu trúc dữ liệu và giải thuật

1. **Thông tin nhóm:**

|  |  |  |
| --- | --- | --- |
| **Họ và tên** | **MSSV** | **Gmail** |
| Võ Trọng Gia Vinh | 18120652 | vgia1055@gmail.com |
| Lê Thành Việt | 18120648 | lethanhviet7c@gmail.com |

**II. Mức độ hoàn thành:**

***1. Những câu đã làm được:***

* + Tìm tất cả công thức đa thức tối thiểu của hàm Bool.
  + Các phép toàn trên vector:
* Phép toán cộng hai vector.
* Phép toán nhân vector với 1 số alpha.
  + Các phép toán trên ma trận:
* Tìm định thức của ma trận.
* Nghịch đảo của ma trận.
* Tích hai ma trận.
* Tìm hạng của ma trận.
* Hệ phương trình tuyến tính.

***2. Những câu chưa làm được:*** 0

**II. Công việc của mỗi thành viên:**

1. **Lê Thành Việt:**

* Tìm công thức tối thiểu của hàm bool:
  + - Nhận input: string BoolExpression::inputBoolExpression()
    - Tách biểu thức thành những phần tử con và biến đổi sang dạng nhị phân: void BoolExpression::creatCell()
    - Xuất kết quả: void BoolExpression::printExpression()
* Các phép toán trên ma trận:
  + - Nhận input: void Matrix::inputMatrix().
    - Tìm định thức của ma trận:
    - double Matrix::determine(double\*\* m,int n).
    - int Matrix::determinantMatrix().
    - Tích hai ma trận: Matrix Matrix::multiplyMatrix(const Matrix& m).
    - Tìm hạng của ma trận: int Matrix::rankMatrix().
    - Xuất ma trận ra màn hình: void showMatrix();

1. **Võ Trọng Gia Vinh:**

* Tìm công thức tối thiểu của hàm bool:
* Xử lí biểu thức dưới dạng nhị phân và tìm công thức tối thiểu: void BoolExpression:: simplifyExpression().
* Tìm công thức rút gọn của biểu thức:

void BoolExpression::compactExpression().

* Các hàm xử lí vector:
* Nhân 2 vector với nhau:

Vector Vector:: operator +( const Vector &a)().

* Nhân vector với số alpha:

Vector Vector:: operator \*(double x)().

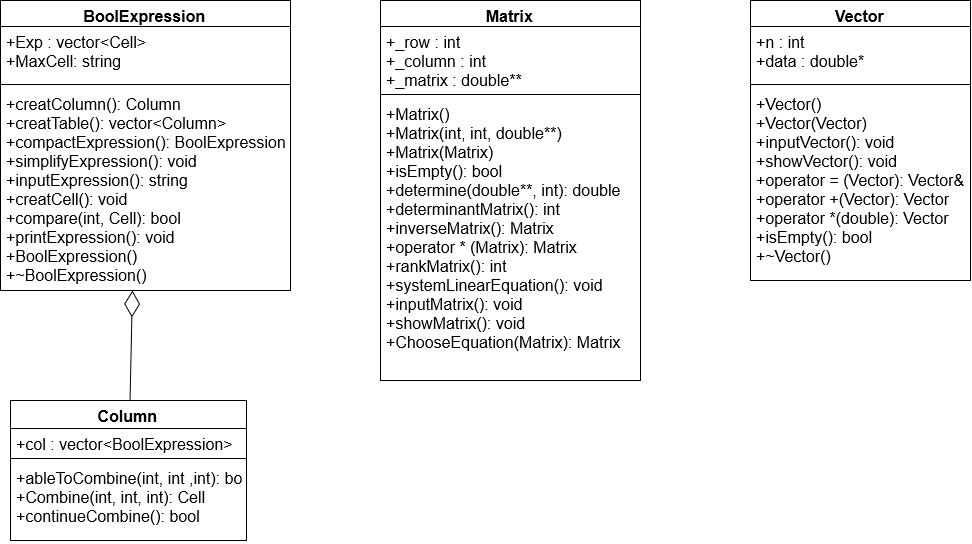
* Các phép toán trên ma trận:
* Giải phương trình tuyến tính:

void Matrix::systemLinerEquation().

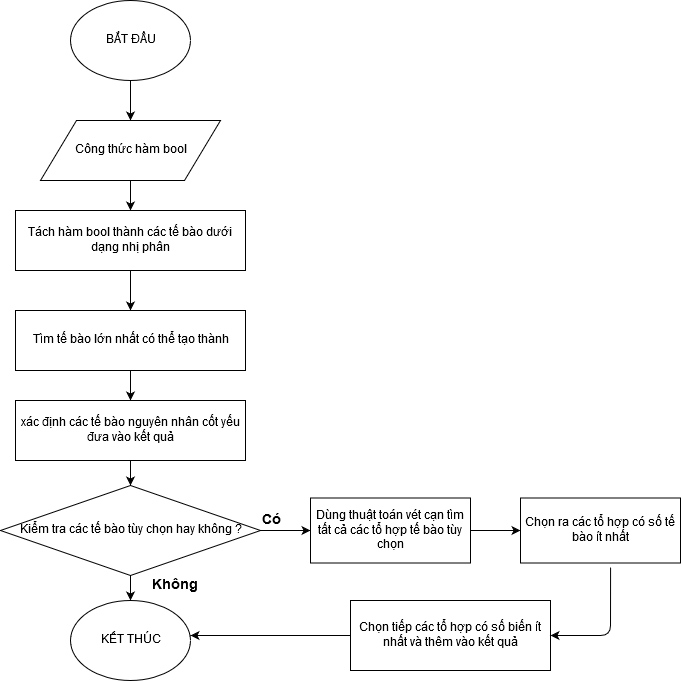
* Tìm ma trận nghịch đảo:

void Matrix::inverseMatrix().

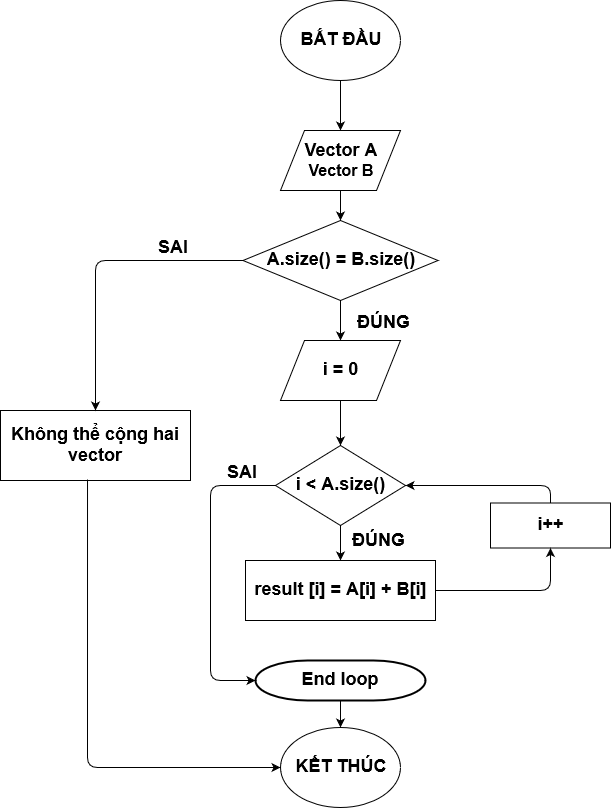
1. **Sơ đồ lớp:**

****

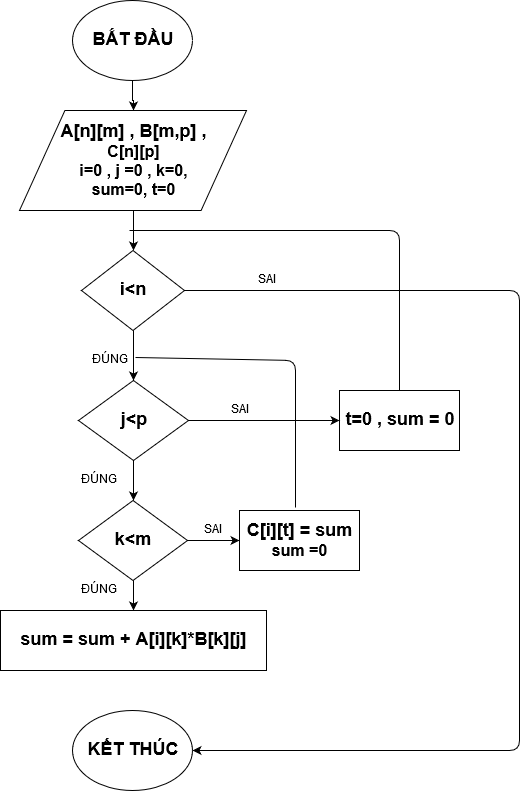
1. **Lưu đồ thuật toán:**
2. *Tìm công thức tối thiểu của hàm Bool:*



1. *Các phép toán trên Vector:*
   * *Cộng hai Vector:*

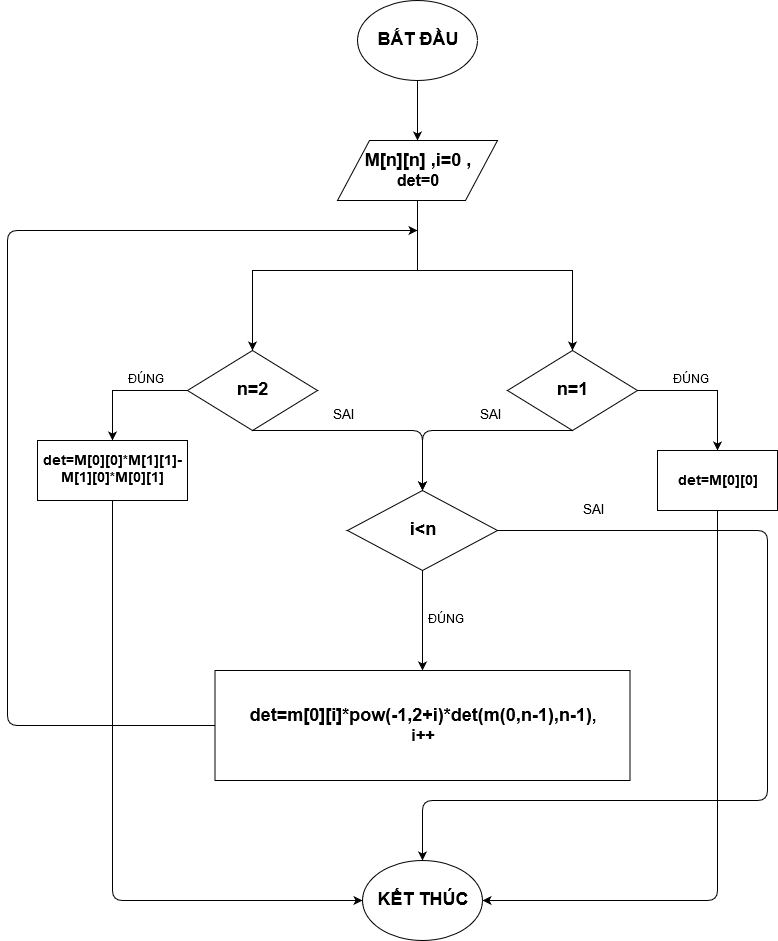


* + *Nhân Vector với một số alpha:*

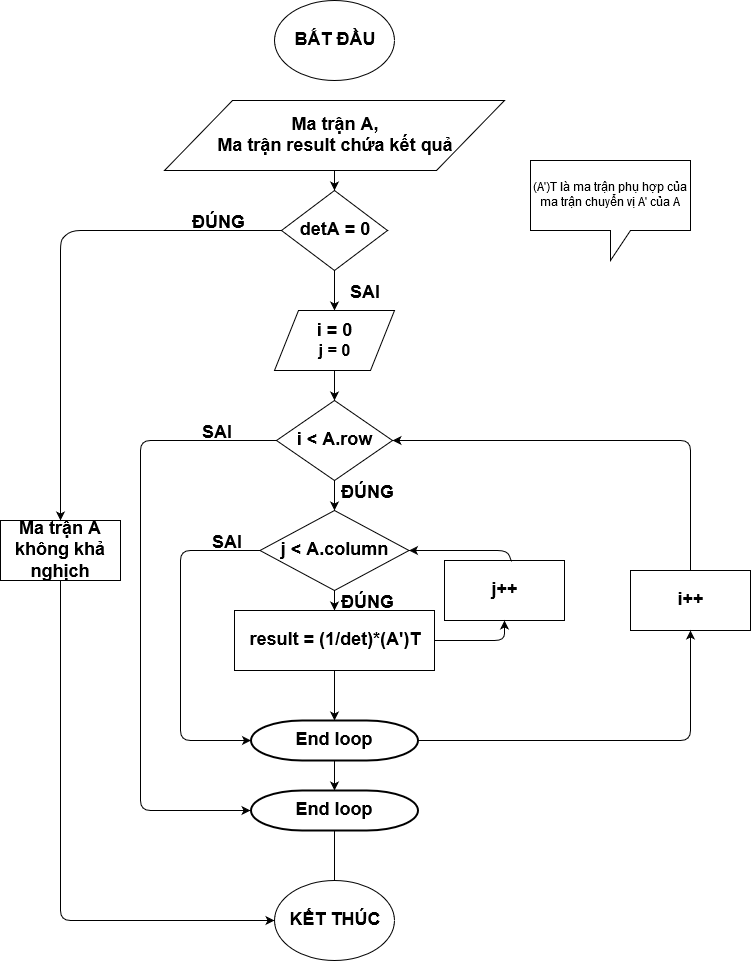


1. *Các phép toán với ma trận:*

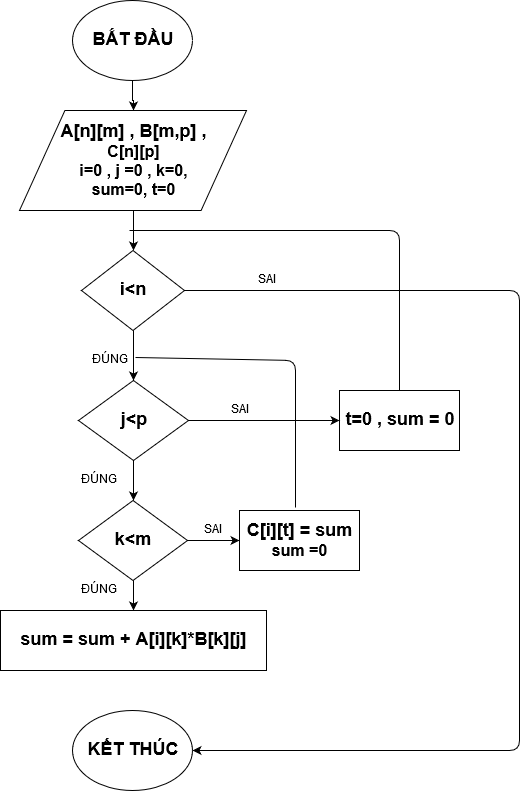
* *Tìm định thức ma trận:*



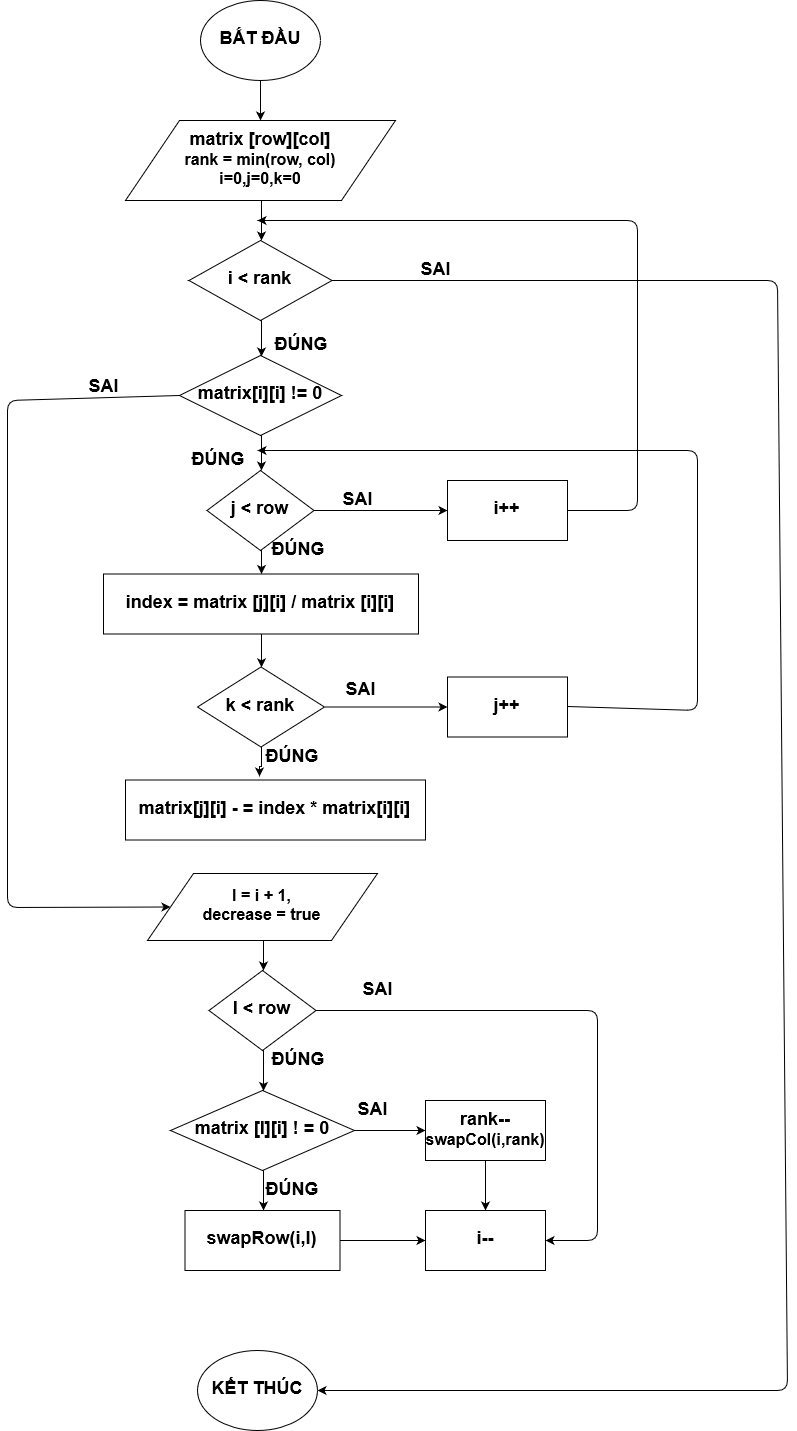
* *Tìm ma trận nghịch đảo:*



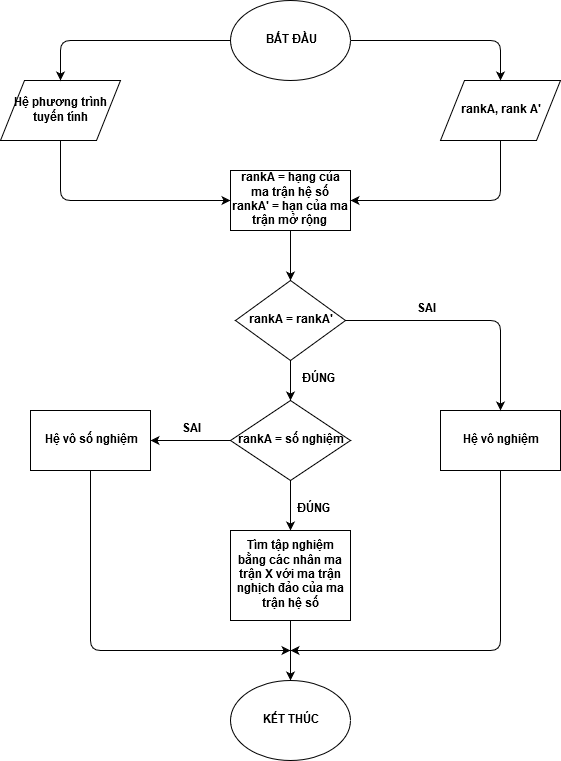
* *Tích hai ma trận:*



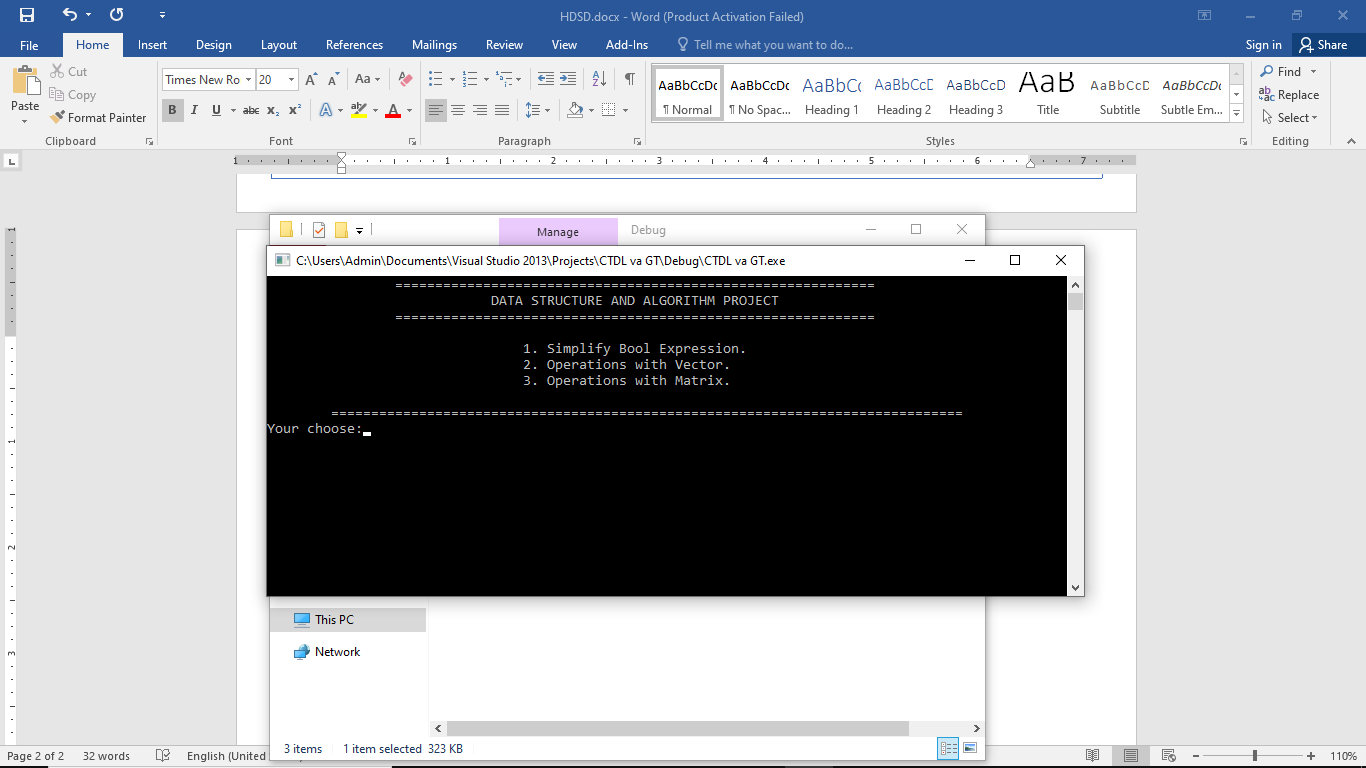
* *Tìm hạng của ma trận:*



* *Hệ phương trình tuyến tính:*



1. **Hướng dẫn sử dụng, hướng dẫn chạy chương trình:**



*Giao diện chương trình*

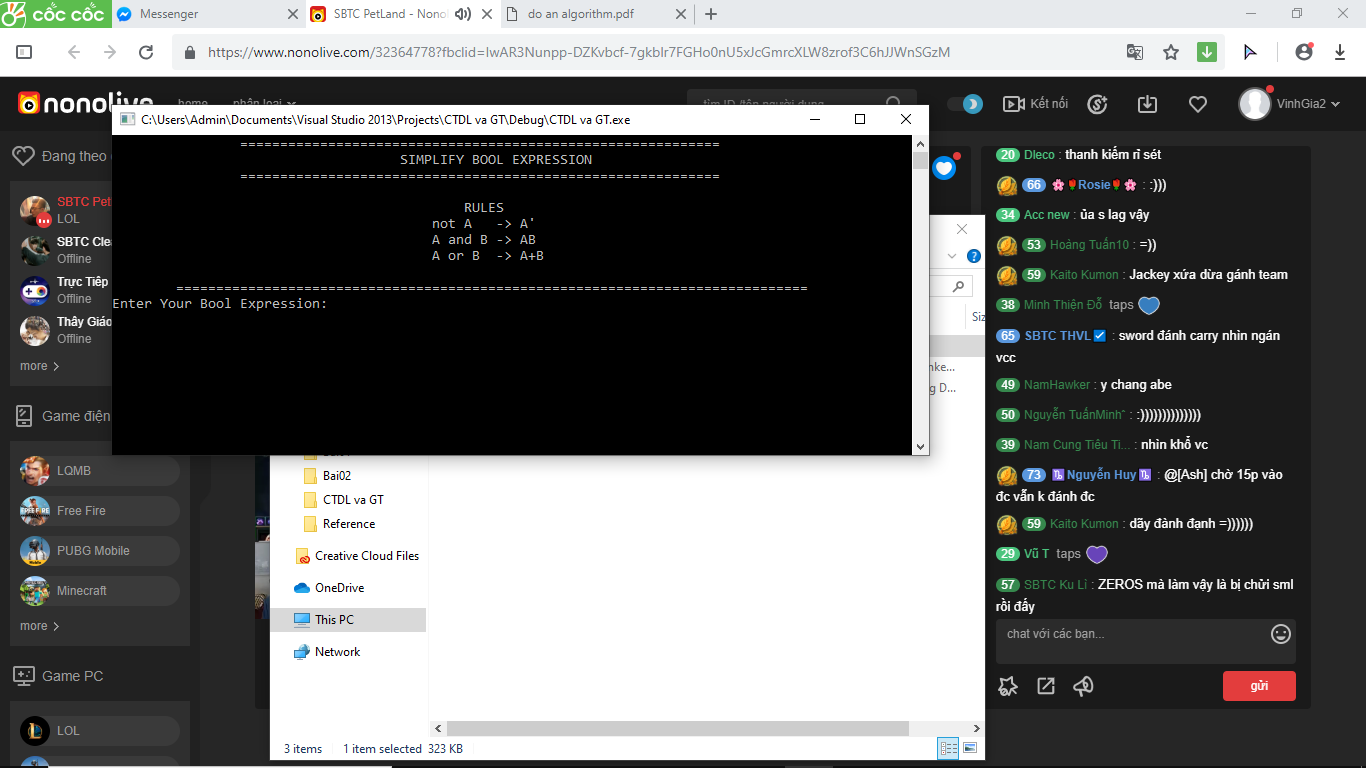
**Trang chính của chương trình bao gồm 3 phần chính:**

1. Rút gọn công thức hàm bool
2. Các phép toán với vector
3. Các phép toán với ma trận

Để lựa chọn chức năng, người dùng nhập vào phần “**Your choose**” số 1, 2 hoặc 3 theo thứ tự ở trên.

Sau khi chọn, chương trình sẽ chuyển sang giao diện của từng chức năng tương ứng.

1. ***Rút gọn công thức hàm bool:***



Tại giao diện này sẽ hiển thị các quy định để nhập công thức hàm bool, *bao gồm:*

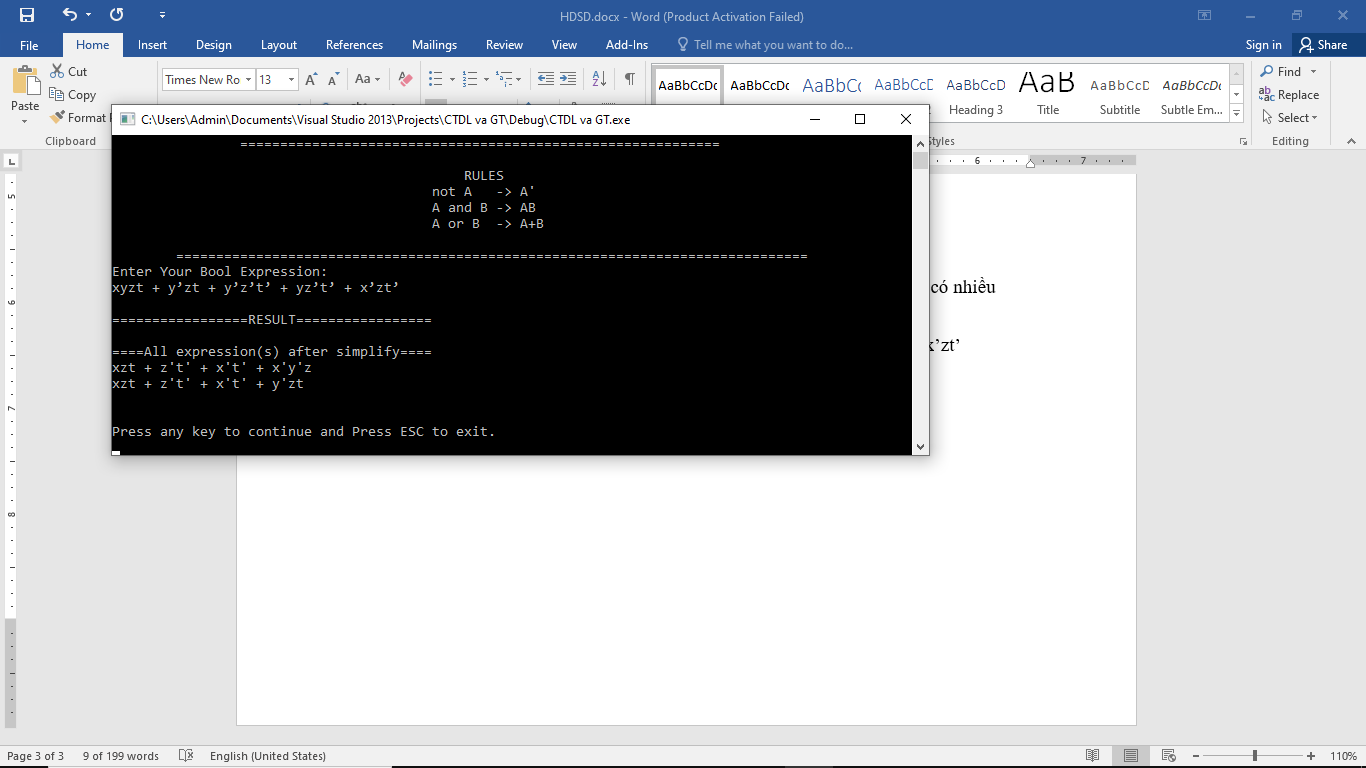
* Not A => A’
* A and B => AB
* A or B => A + B

***Lưu ý:*** các biến phân biệt chữ hoa và chữ thường.

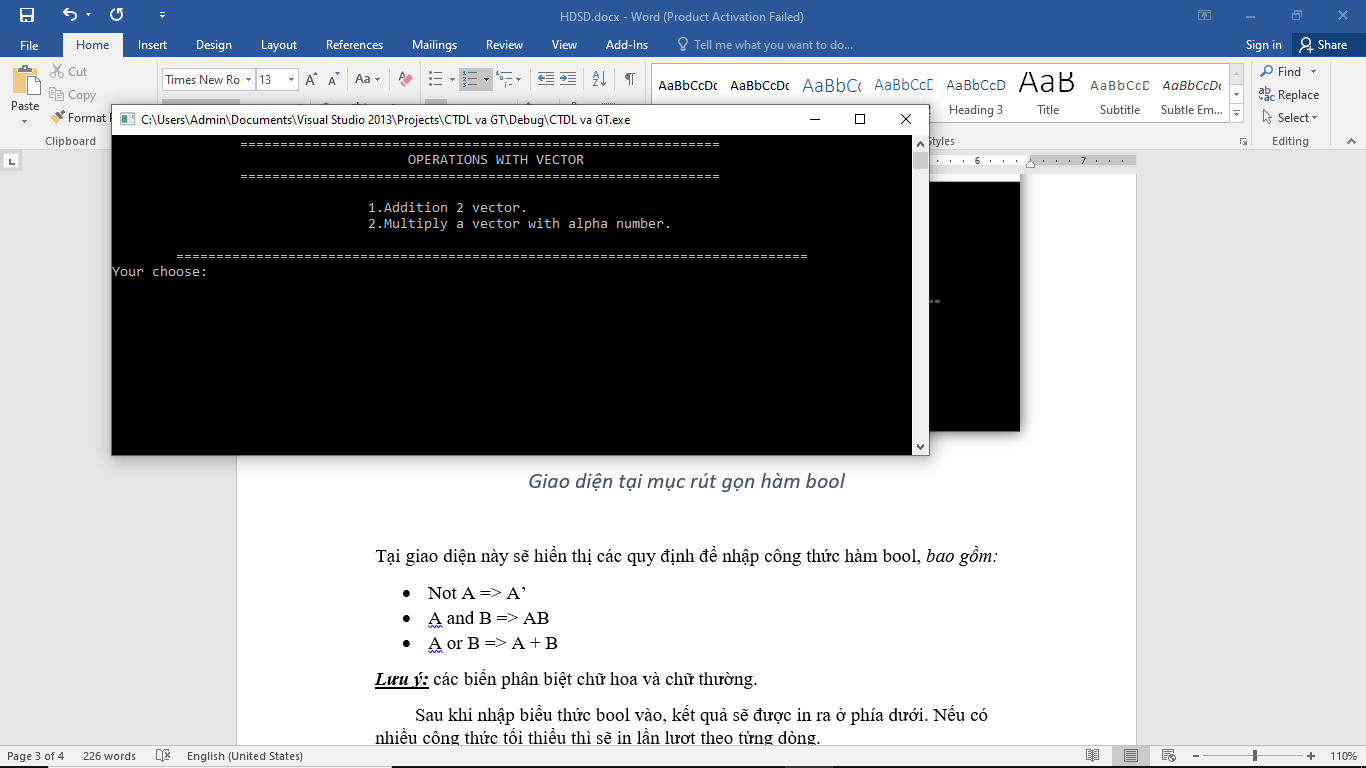
Sau khi nhập biểu thức bool vào, kết quả sẽ được in ra ở phía dưới. Nếu có nhiều công thức tối thiểu thì sẽ in lần lượt theo từng dòng.

Sau khi rút gọn xong thì nhấn nút bất kì để trở về trang chính, hoặc ESC để thoát chương trình.

***Ví dụ:*** Rút gọn công thức tối thiểu hàm f = xyzt + y’zt + y’z’t’ + yz’t’ + x’zt’



1. ***Các phép toán với vector:***



Tại giao diện này, ta có 2 lựa chọn:

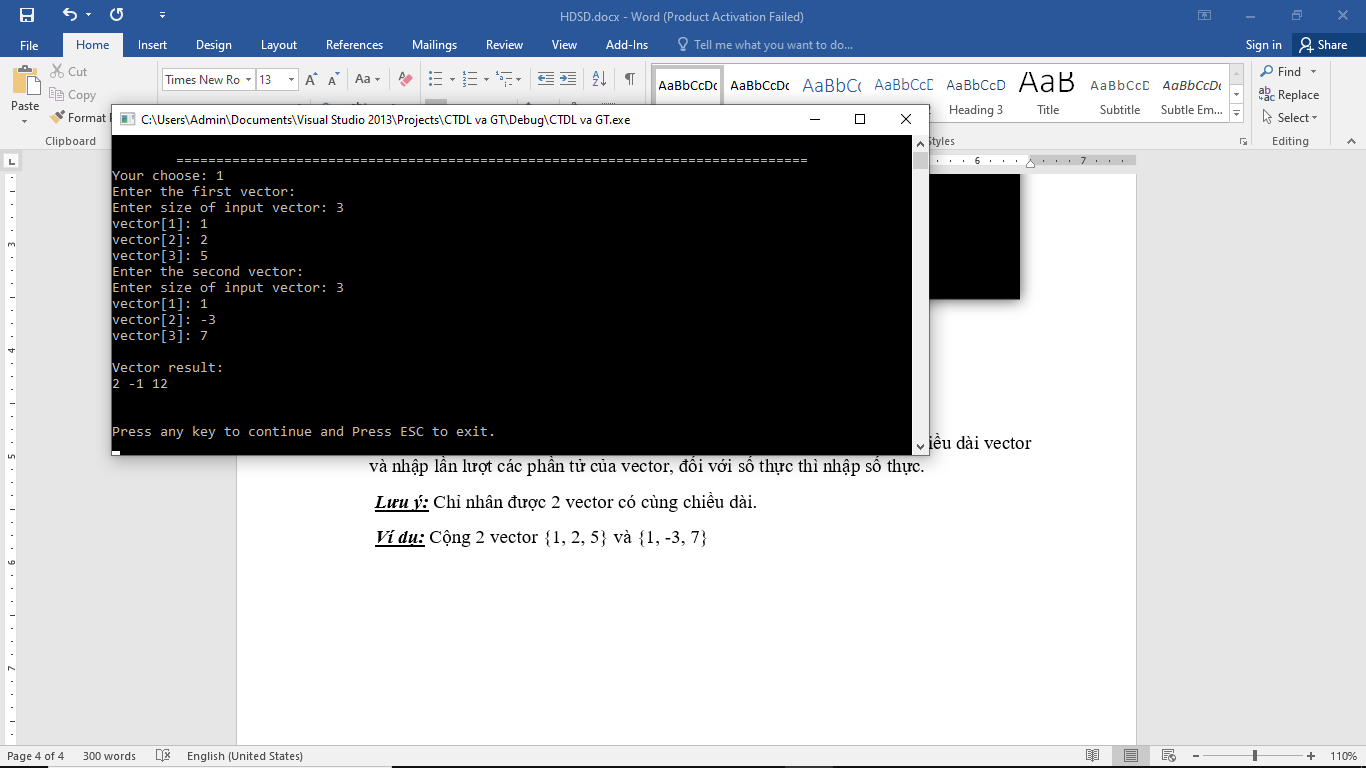
1. Cộng 2 vector với nhau
2. Nhân một vector với một số thực alpha

Sau khi chọn thì ta nhập input theo thứ tự, đối với vector thì nhập chiều dài vector và nhập lần lượt các phần tử của vector, đối với số thực thì nhập số thực.

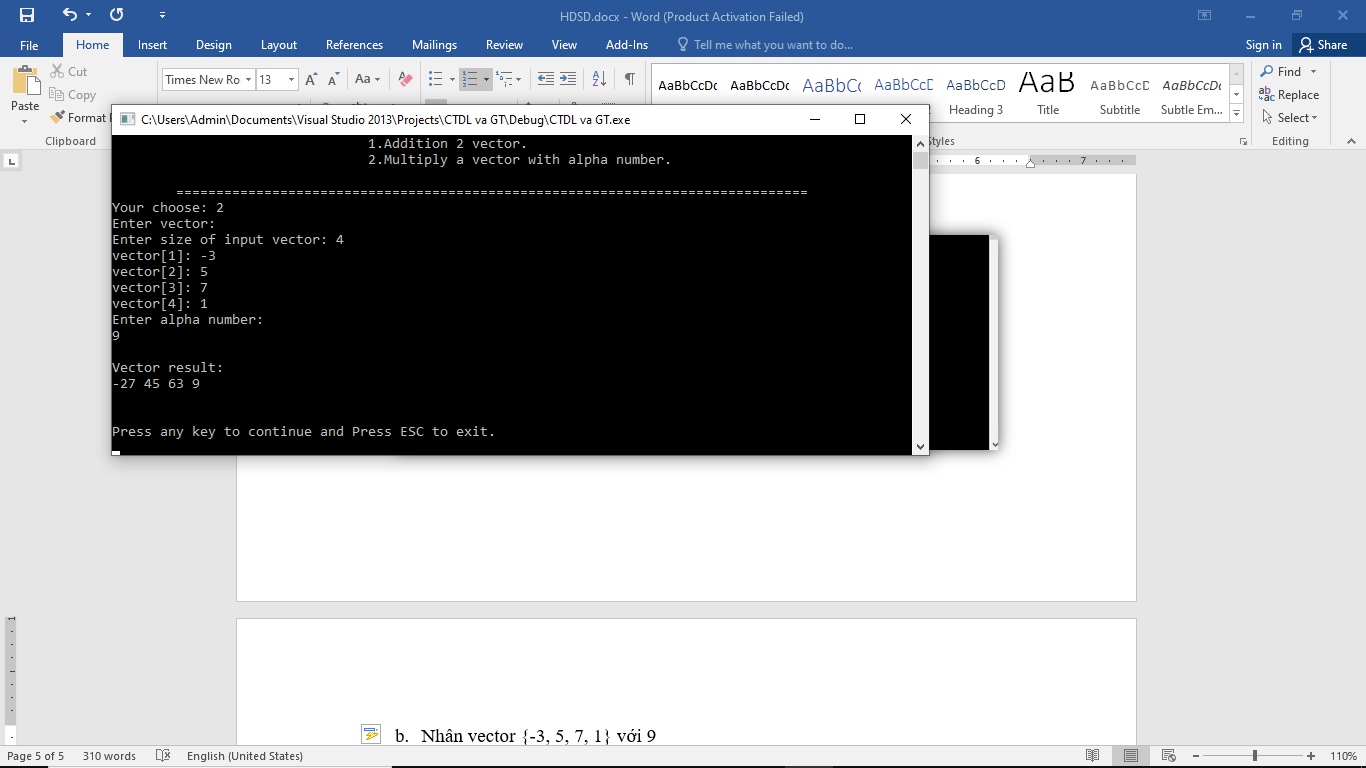
***Lưu ý:*** Chỉ nhân được 2 vector có cùng chiều dài.

***Ví dụ:***

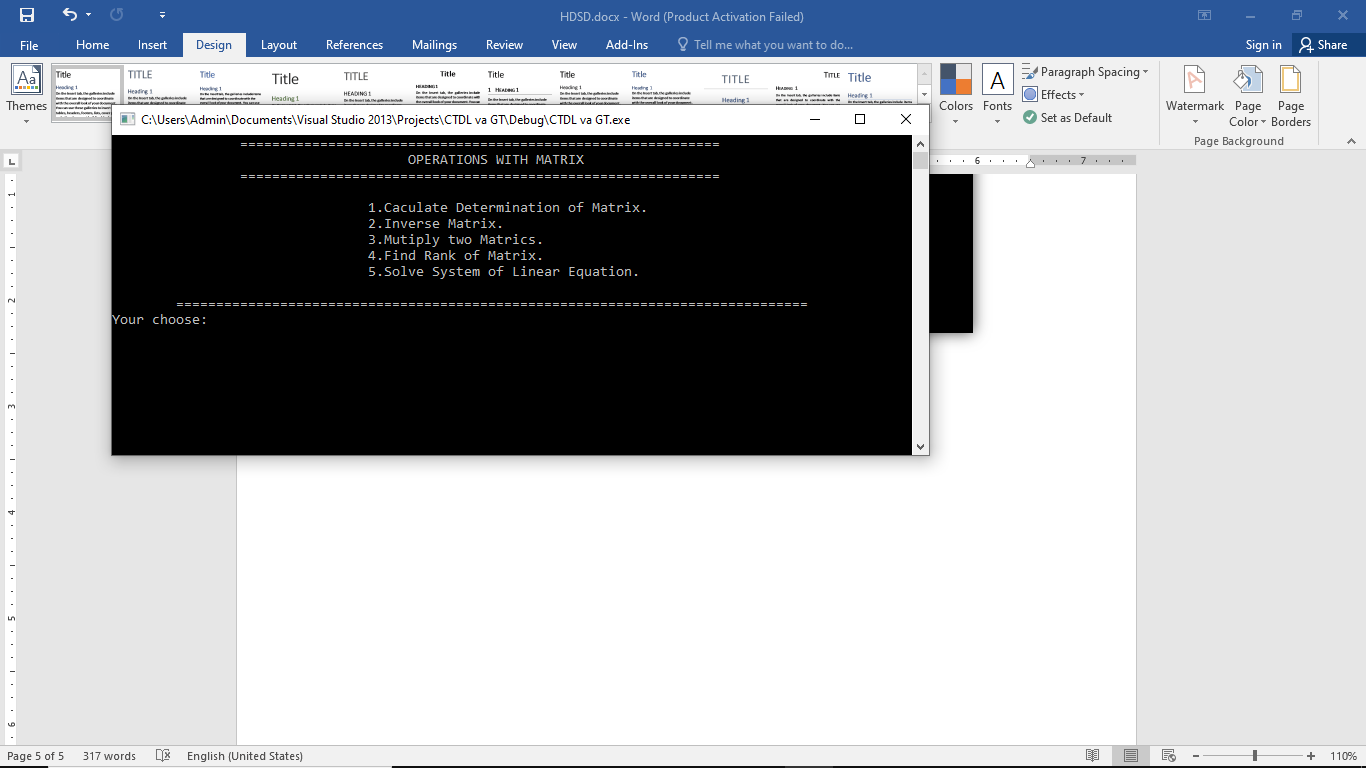
1. Cộng 2 vector {1, 2, 5} và {1, -3, 7}



1. Nhân vector {-3, 5, 7, 1} với 9



1. ***Các phép toán với ma trận:***



Ở mục này ta có 5 lựa chọn:

1. Tính định thức của ma trận
2. Tìm ma trận nghịch đảo
3. Nhân 2 ma trận với nhau
4. Tìm hạng của ma trận
5. Giải hệ phương trình tuyến tính

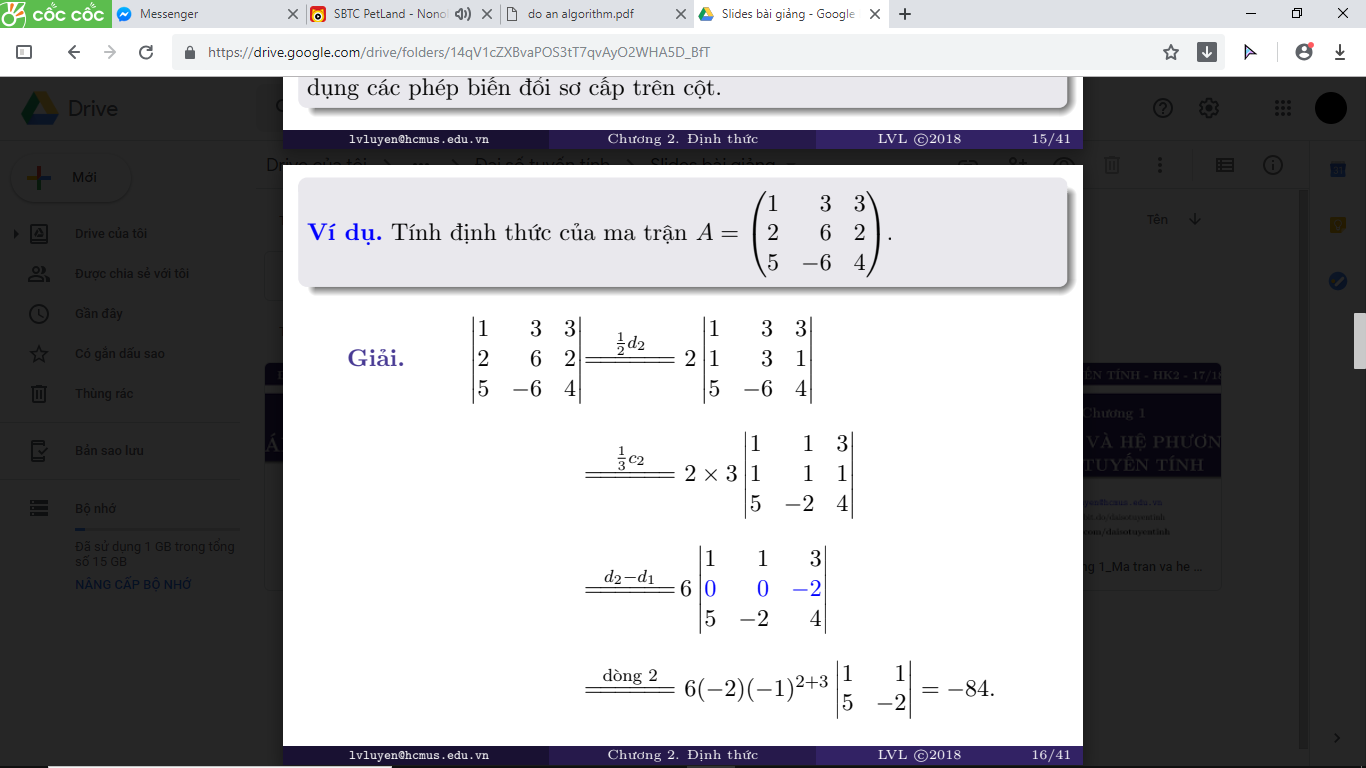
Sau khi chọn thì ta nhập input theo thứ tự, đối với ma trận thì nhập lần lượt số dòng, số cột sau đó là lần lượt nhập các phần tử theo từng dòng

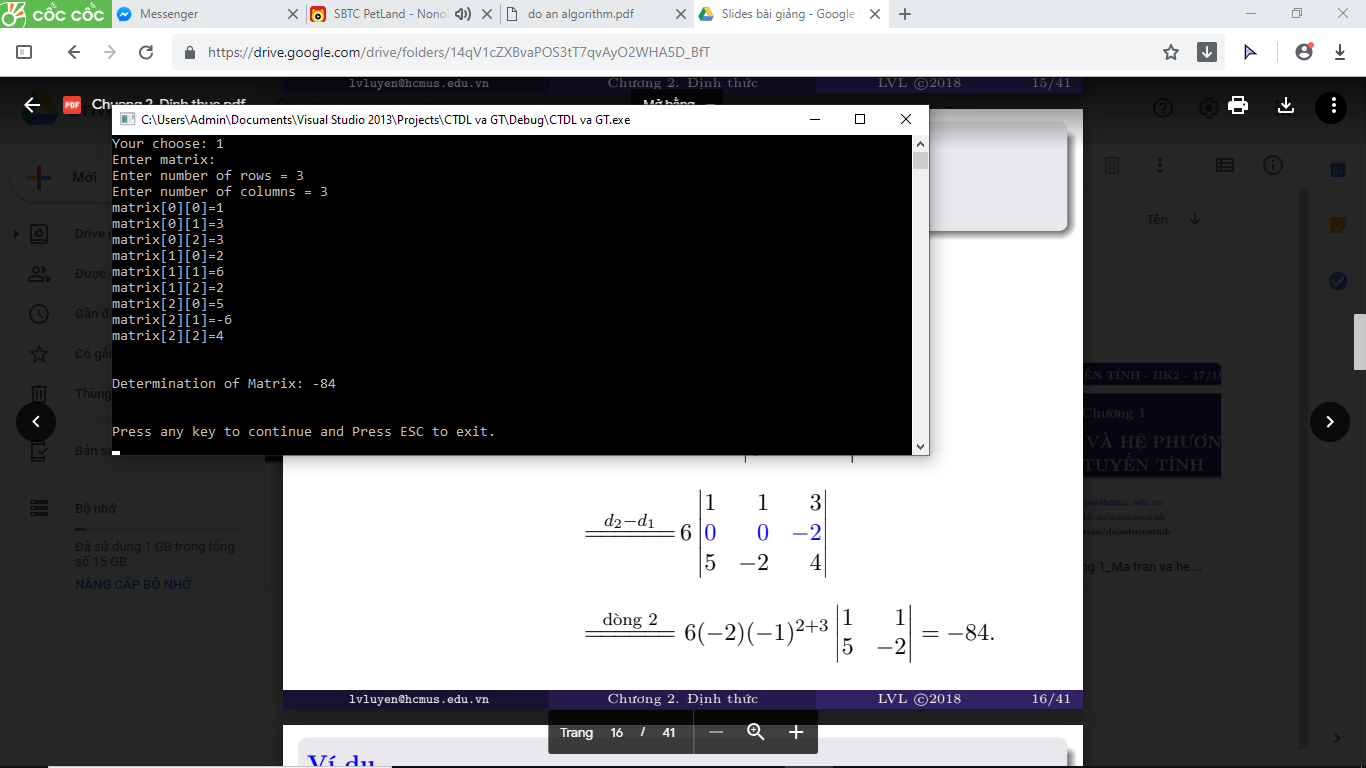
Đối với hàm giải hệ phương trình tuyến tính thì ta sẽ nhập vào số ẩn và số phương trình, sau đó ta sẽ nhập ma trận có kích thước (số phương trình, số ẩn + 1). Nếu phương trình *vô nghiệm* thì chương trình sẽ in ra “The equation has no solution!!!”, nếu có *vô số nghiệm* sẽ in ra “The equation has countless solutions!!!”, nếu có đúng 1 nghiệm sẽ in ra theo dạng {x1, x2, …, xn}

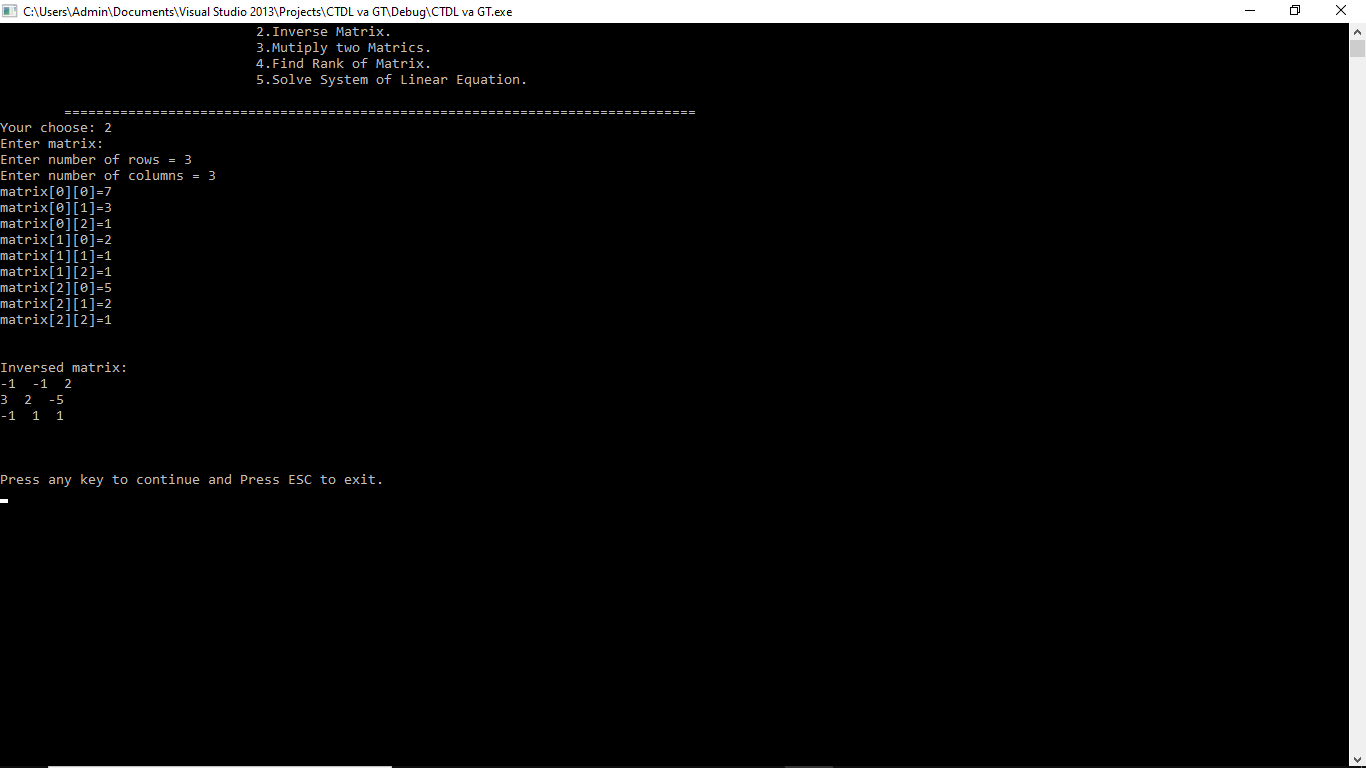
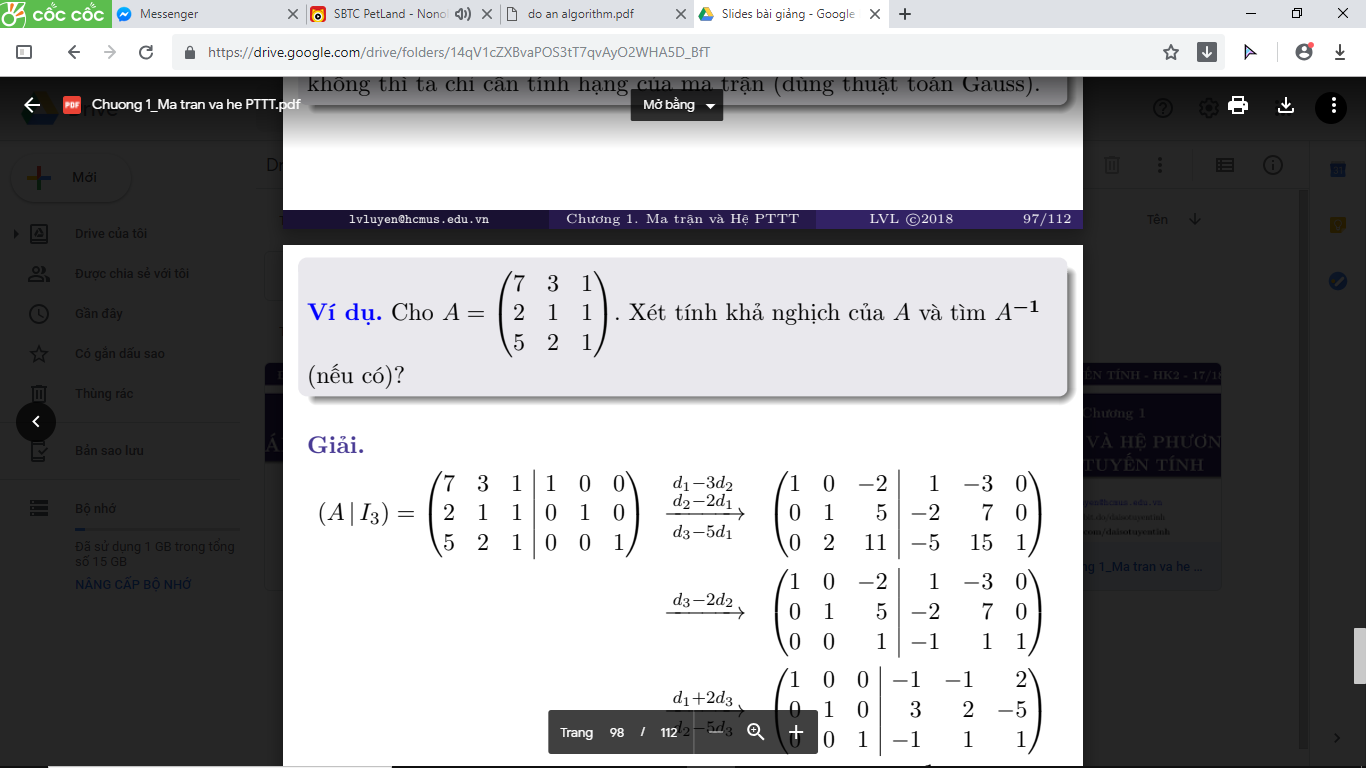
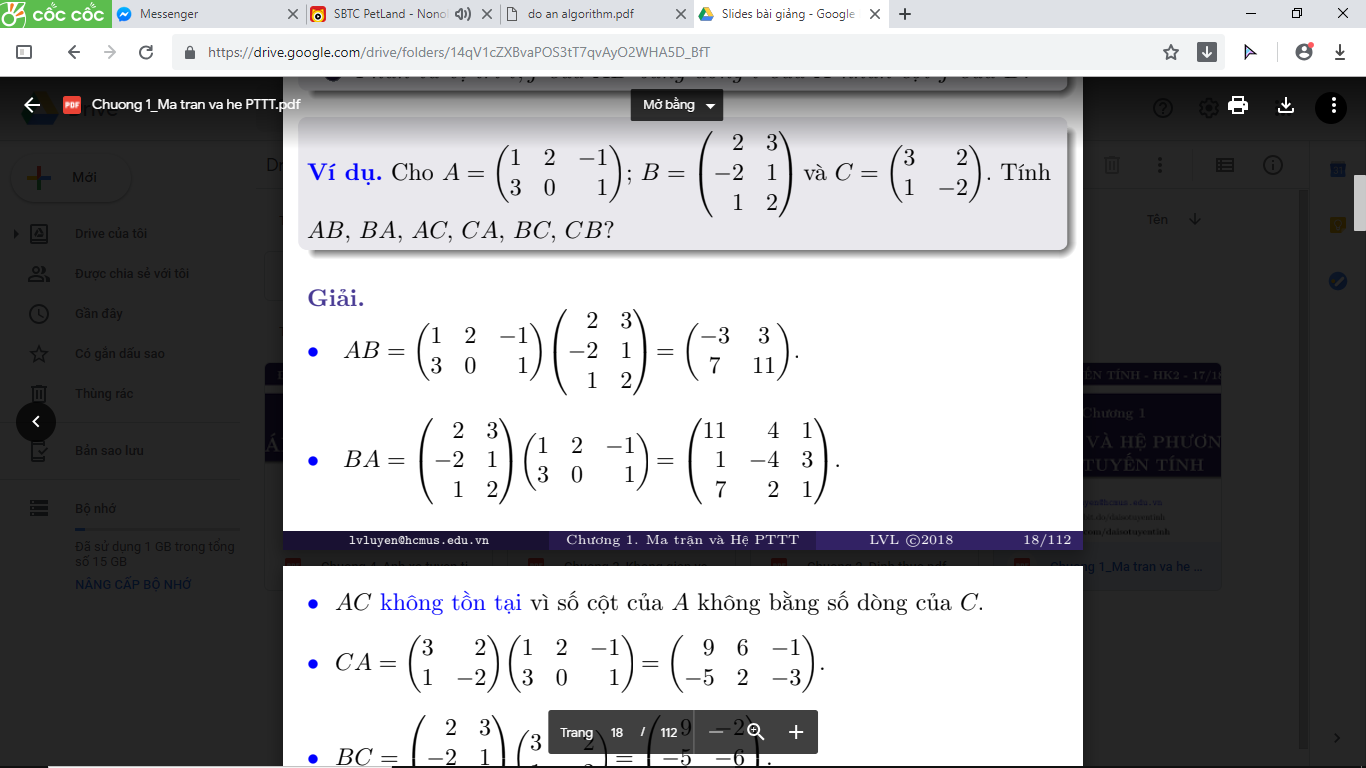
***Lưu ý:***

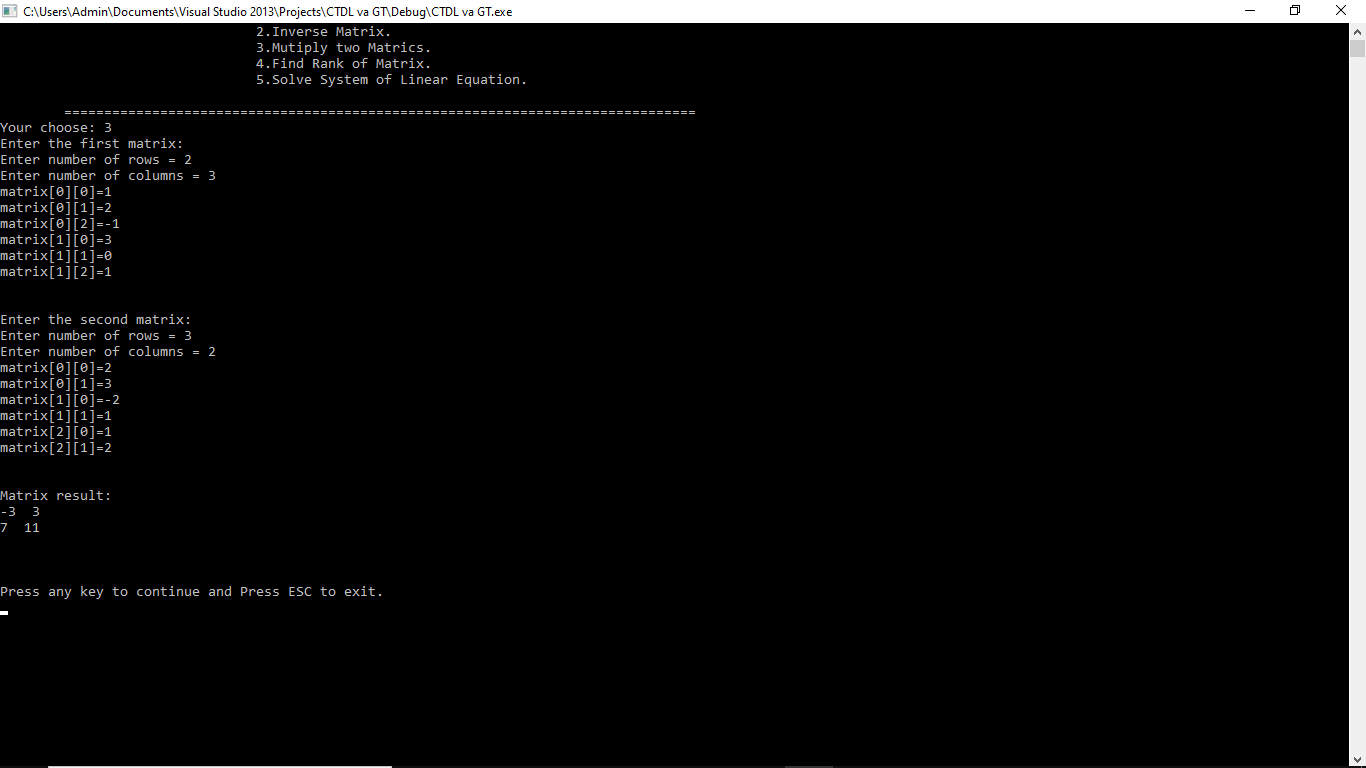
* Chỉ có thể tính định thức và tìm ma trận nghịch đảo của ma trận vuông
* 2 ma trận A và B muốn nhân được với nhau thì ***số cột của ma trận A*** phải bằng với ***số dòng của ma trận B***

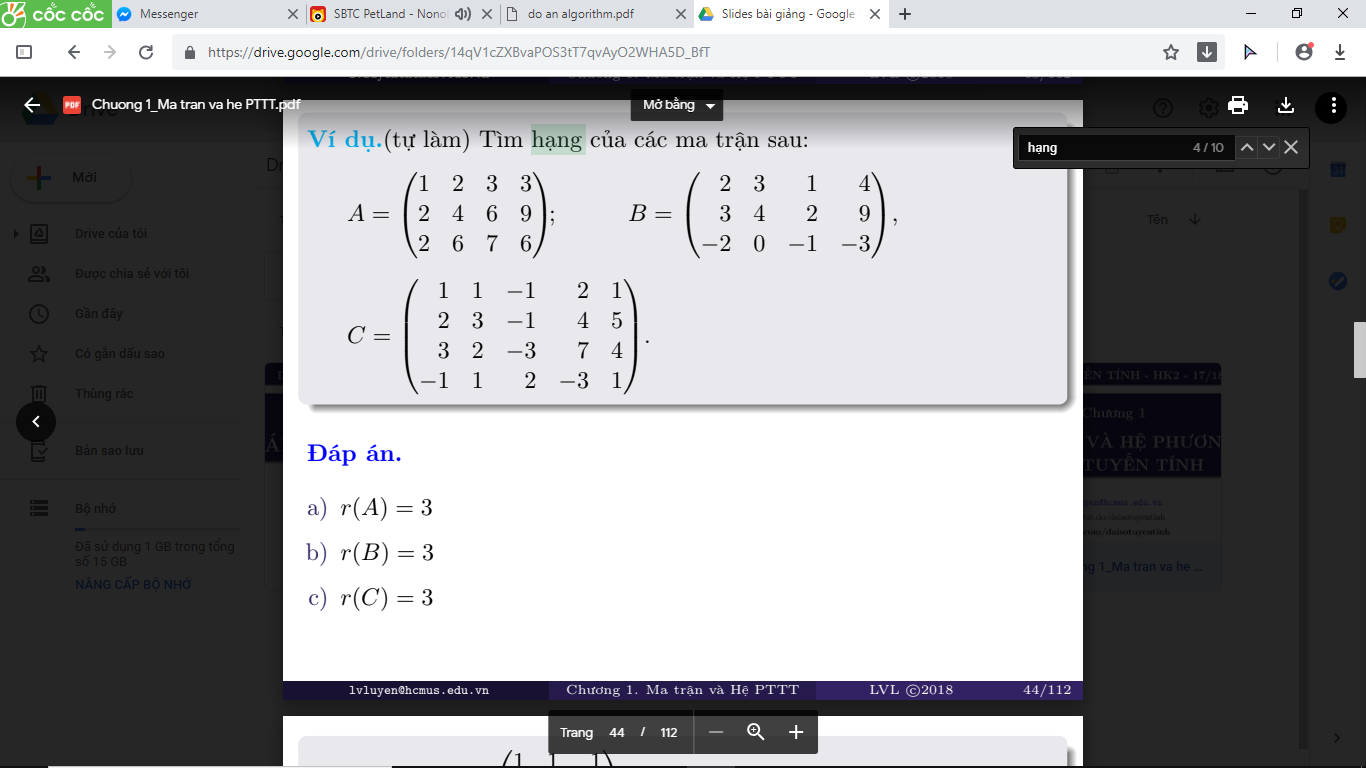
***Ví dụ:***

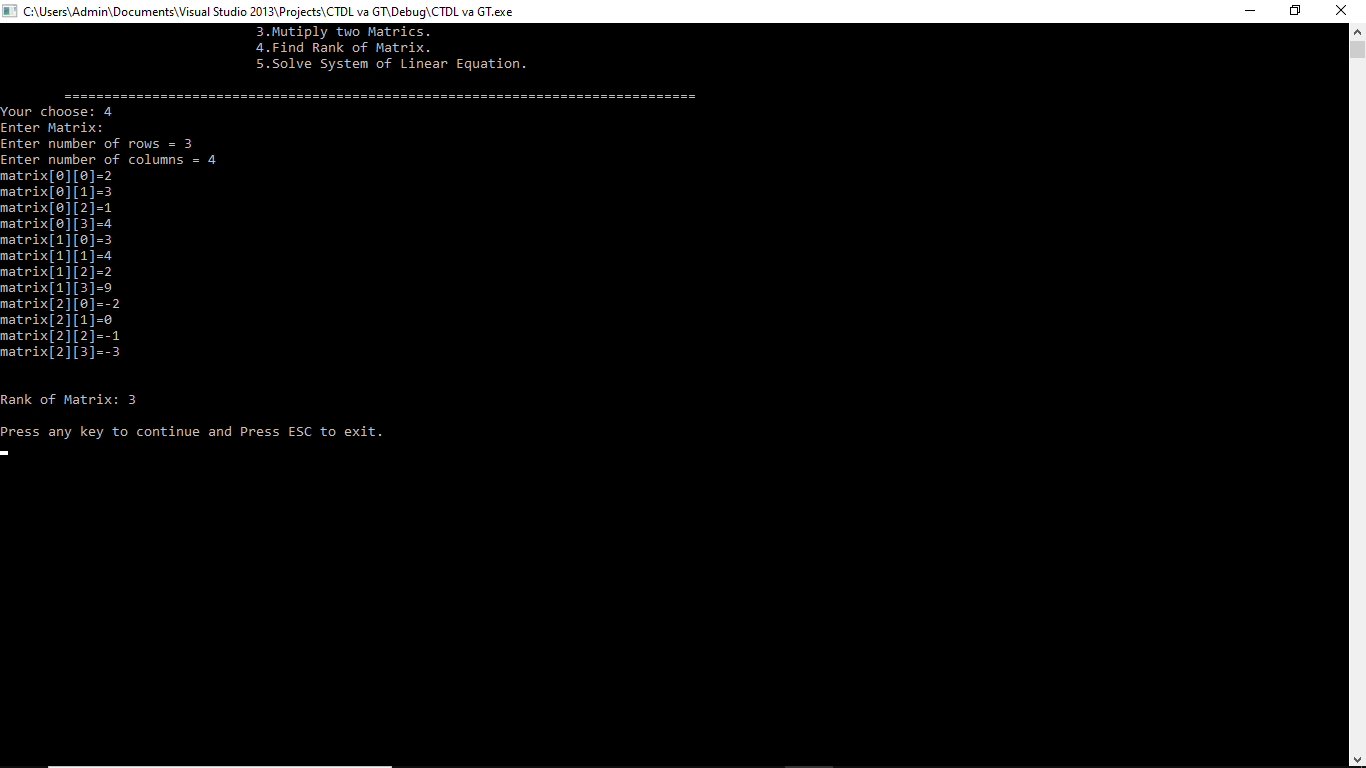
1. Tìm định thức của ma trận 

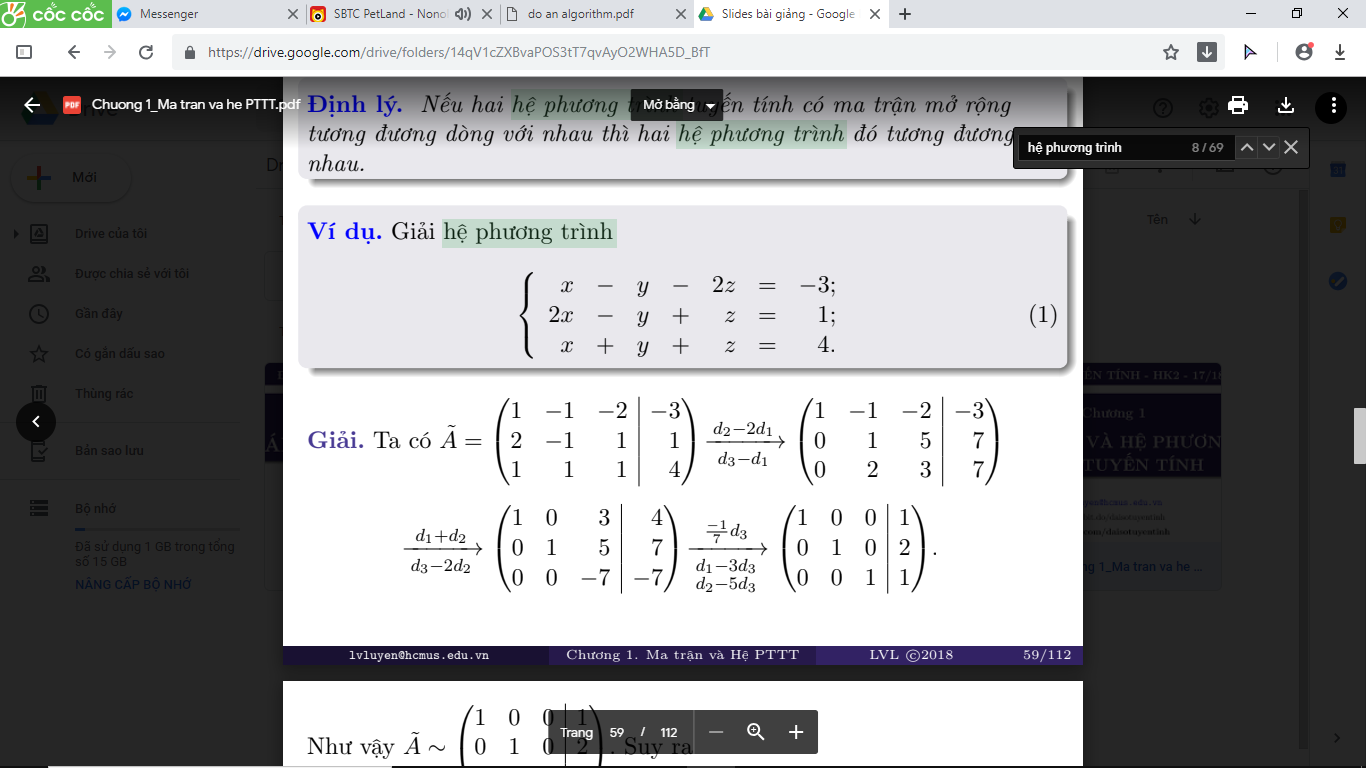


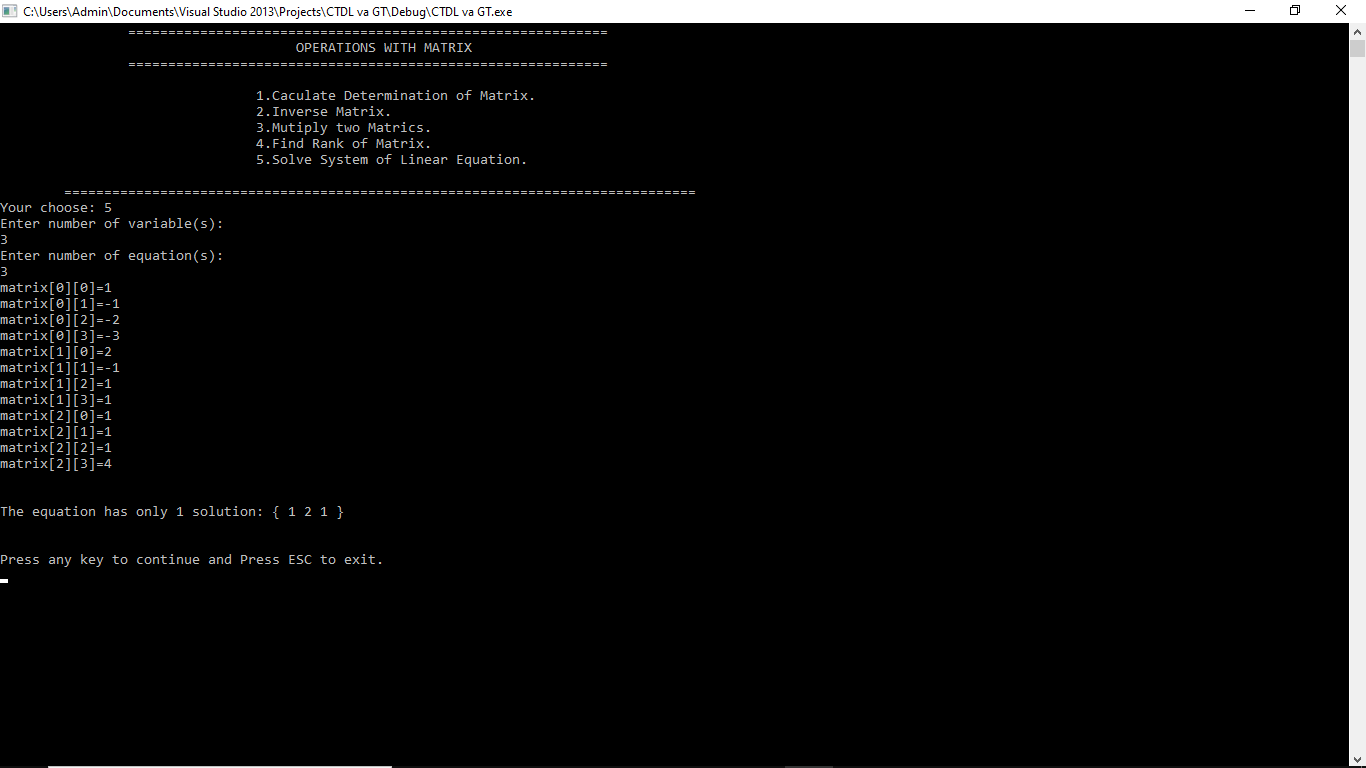
1. Tìm ma trận khả nghịch của ma trận 
2. Nhân hai ma trận 



1. Tìm hạng của ma trận 



1. Giải hệ phương trình 



1. **Video clip hướng dẫn sử dụng:**

[Link video !](https://drive.google.com/file/d/1dTS43BVk5iXE5mAXJWOWcu-0JX2fk56D/view?usp=sharing) ( Mở chất lượng cao để xem rõ hơn )