ĐẠI HỌC KHOA HỌC TỰ NHIÊN- ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN

**Báo cáo đồ án:**

**THỰC HÀNH CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT**



Họ và tên:Nguyễn Nhị Cẩm Tiên

MSSV:18120588

**Chưa làm được:**

-Hàm bool

Đã làm được:

-Vector: cộng 2 vector và nhân vector cho một số k

-Matrix: tính tích 2 ma trận, định thức ma trận, ma trận nghịch đảo, hạng ma trận, hệ phương trình tuyến tính

**1.Vector:**

Xây dựng class vector gồm 2 thành phần: một mảng một chiều lưu giá trị vector(valueVT) và biến n lưu chiều dài vector.

Class gồm các hàm:

**Hàm tạo:**

Vector();

Vector(int N, double x);

Vector(const Vector &a);

**Hàm huỷ**

~Vector();

**Hàm nhập xuất:**

void input();// enter vector

void output();//print vector

**Hàm cộng 2 vector:**

int add(const Vector &a);

Nếu số phần tử của 2 vector khác nhau return 0, ngược lại thực hiện cộng lần lượt các phần tử của 2 vector có cùng chỉ số.

**Hàm nhân vector với một số:**

void multiplyK(const double &k);

Chạy vector từ đầu tới cuối, nhân các phần tử cho k.

**2.Matrix**

Xây dựng class Matrix gồm 3 thành phần: m dòng, n cột và valueMT để lưu giá trị các phần tử.

**Hàm tạo:**

Matrix();

Matrix(const Matrix & a);

**Hàm huỷ:**

~Matrix();//delete matrix

**Hàm nhập xuất:**

void Input();//enter matrix

void OutPut();//print matrix

**Hàm tính tích 2 phần tử:**

int Multiply(const Matrix & a);//return 1 if if possible multiply else return 0

Nếu chỉ số cột và dòng của 2 matrix khác nhau thì return 0;

Tạo một mảng temp 2 chiều có chỉ số dòng và cột tương đương

Giá trị các phần tử mảng temp là temp[i][j] = temp[i][j] + valueMT[i][k] \* a.valueMT[k][j]

**Hàm tính định thức:**

double determinant();// return -99 if can not calculate determinant else return determinant

Hàm trả về -99 nếu không phải ma trận vuông.

Để xây dựng hàm “determinant” ta xây dựng hàm “det” là hàm tính định thức ma trận vuông.

Hàm det:Ta xây dựng mảng temp là mảng giá trị Matrix đang tính định thức bỏ đi đường chéo chính rồi tính theo công thức:

d = d + (pow(-1, c) \* matrix[0][c] \* det(n - 1, temp));(sử dụng thuật toán đệ quy)

**Hàm tìm ma trận nghịch đảo(chỉ áp dụng cho ma trận vuông):**

void Cofactor(double\*\*&temp, int p, int q, int n);: lấy matrix temp bỏ đi hàng p và cột q// ma trận phù hơp

double\*\* adjoint();: tại một vị trí ta lấy tổng chỉ số hàng i và cột j. Nếu tổng chia hết cho 2 thì lấy k=1 ngược lại k=-1. Giá trị từng vị trí trong mảng là định thức của cofactor của vị trí đó nhân với k.//ma trận chuyển vị

void Inverse(double deter);: giá trị từng phần tử là adjoint ở vị trí tương ứng chia cho định thức của ma trận.

**Hàm tìm hạng ma trận:**

int Rank();

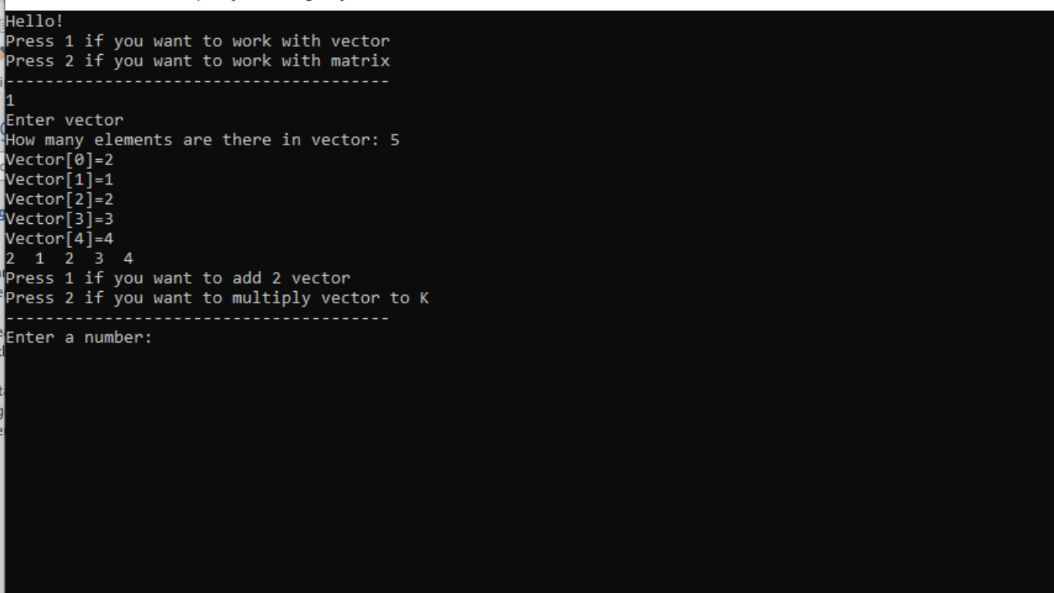
Ta lấy biến rank=số dòng của ma trận.

Nếu số cột=1 thì ma trận có rank=1

Xét một hàng i với giá trị từ đầu của nó tới giá trị ở vị trí (i,i)=0

Hướng dẫn sử dụng bằng hình ảnh:

Bước đầu vào file .exe, nhấn 1 nếu muốn tính toán vector, nhấn 2 nếu muốn tính toán ma trận.

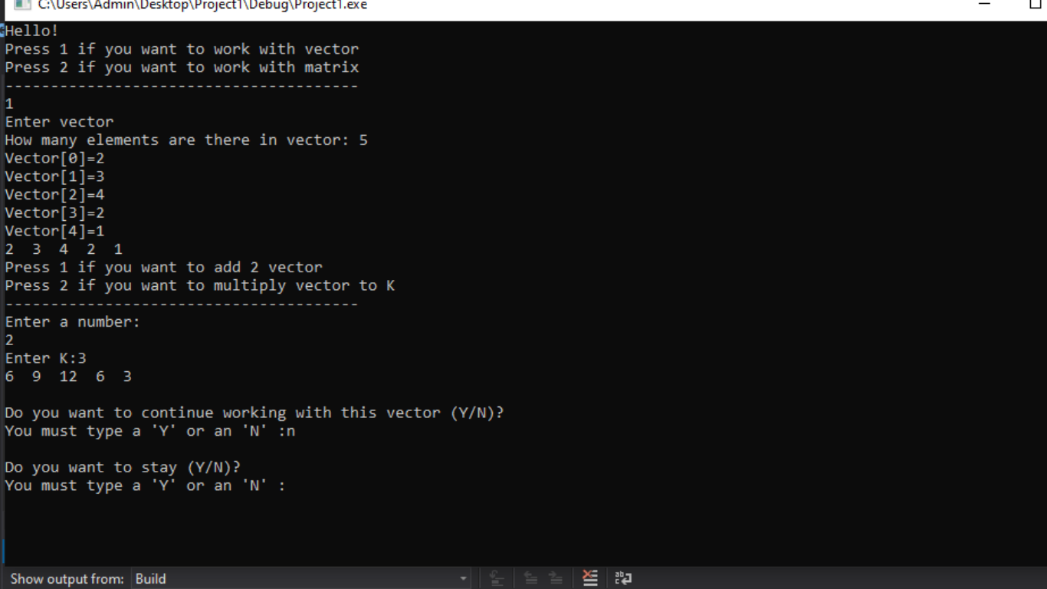


Nếu nhấn 1:

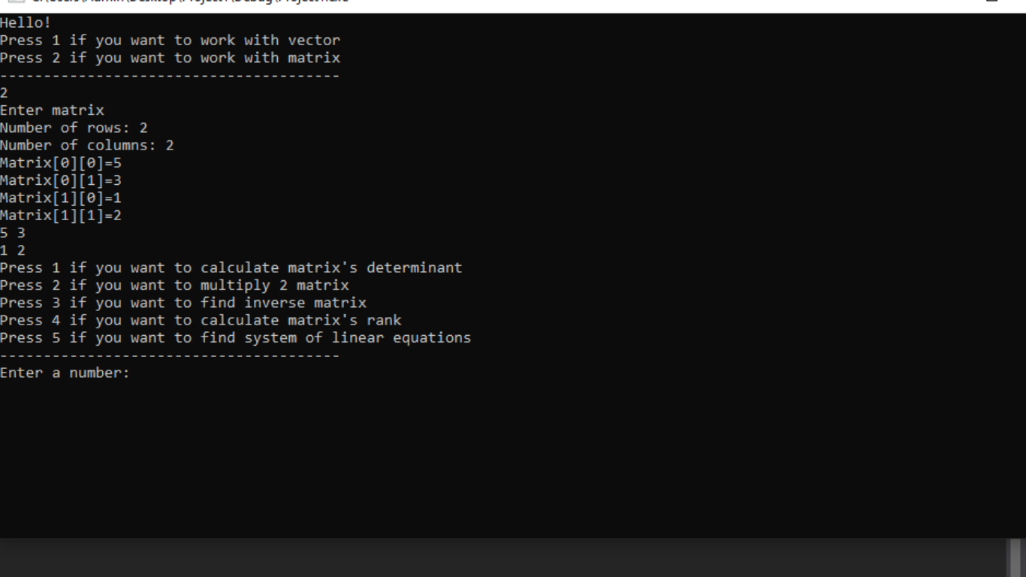
Nhập vào số phần tử của vector và nhập các giá trị phần tử, nhấn 1 nếu bạn muốn cộng 2 vector, nhấn 2 nếu muốn nhân vector cho một số k.

Sau khi kết thúc, chương trình sẽ hỏi bạn có muốn tiếp tục làm việc với vetor này không, nhấn Y hoặc y nếu muốn, nếu không thì bạn nhấn N hoặc n.

Nếu bạn không muốn tiếp tục làm việc với vector này, chương trình sẽ hỏi bạn có muốn ở lại(làm việc với matrix hoặc với một vector khác), nhấn y hoặc n để lựa chọn



Làm việc với matrix: đầu tiên nhập số hàng và số cột, sau đó lần lượt nhập các giá trị phần tử cho matrix.

1. Nhấn 1 để tính định thức
2. Nhấn 2 để nhân ma trận với một ma trận khác
3. Nhấn 3 để tìm ma trận nghịch đảo
4. Nhấn 4 để tính hạng ma trận
5. Nhấn 5 để tìm hệ phương trình tuyến tính.

Các câu hỏi khi kết thúc sẽ tương tự với khi làm việc cùng vector:

