KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
227 Nguyễn Văn Cừ, Phường 4, Quận 5, TP.HCM
Điện Thoại: (08) 38.354.266 - Fax:(08) 38.350.096

BÁO CÁO CÁCH TÌM MA TRẬN NGHỊCH ĐẢO BẰNG GIẢI THUẬT GAUSS-JODAN BẰNG NGÔN NGỮ PYTHON

Name: Nguyễn Trọng Kha

ID: 19120083

I.Ý tưởng chính của chương trình:

-Thông qua giải thuật gauss-jodan:

+Bài làm sẽ tạo ra 2 ma trận song song là ma trận cần tìm và ma trận đơn vị ứng với ma trận cần tìm.

+Sau đó sẽ tiến hành biến đổi các dòng và các cột sau cho biến thành ma trận bậc thang rút gọn từ đó ta thu được ma trận nghịch đảo

-Bên cạnh đó sẽ có các dòng code kiểm tra xem ma trận cần tìm có ma trận nghịch đảo hay không

II.Ý nghĩa các hàm quan trọng:

```
def Tao_Ma_Tran_Don_Vi(A):
    C = copy.deepcopy(A)
    n = len(A)
    for i in range(0, n):
        for j in range(0, n):
            if i == j:
                C[i][j] = 1
            else:
                C[i][j] = 0
    return C
```

-Hàm def Tao_Ma_Tran_Don_Vi(A) có vai trò tạo ra Ma trận đơn vị ứng với ma trận ban đầu mình nhập vào

```
def Is_Bang(A, B):
    n = len(A)
    for i in range(0, n):
        for j in range(0, n):
            if A[i][j] != B[i][j]:
                return False
    return True


def Mul(A, B):
```

```
    n = len(A)
    C = []
    for i in range(0, n):
        C.append([])
        for j in range(0, n):
            C[i].append(0)
    for i in range(0, n):
        for j in range(0, n):
            C[i][j]=0
            for k in range(0, n):
                C[i][j]+=A[i][k]*B[k][j]

    return C
```

Hàm def Is_Bang(A, B) có vai trò kiểm tra xem 2 ma trận có bằng nhau không và hàm def Mul(A, B) để nhân 2 ma trận

```
def Tim_Ma_Tran_Nghich_Dao(A):
    R = copy.deepcopy(A)
    C=Tao_Ma_Tran_Don_Vi(A)
    Z = Tao_Ma_Tran_Don_Vi(A)
    m, n = len(R), len(R[0])
    row = col = 0
    while row < m:
        while col < n and all(is_zero(R[i][col]) for i in range(row, m)):
            col += 1
        if col == n: break
        pivot_row = row + [is_zero(R[i][col]) for i in range(row, m)].index(False)
        row_switch(R, row, pivot_row)
        row_switch(Z, row, pivot_row)
        t = 1 / R[row][col]
        row_mul(R, row, 1 / R[row][col])
        row_mul(Z, row, t)
        for i in range(row + 1, m):
            t2 = -R[i][col]
            row_add(R, i, row, -R[i][col])
            row_add(Z, i, row, t2)
        row += 1
    for i in range(n - 2, -1, -1):
```

```
        for j in range(i + 1, n):
            t3 = -R[i][j]
            row_add(R, i, j, -R[i][j])
            row_add(Z, i, j, t3)
    if Is_Bang(C,Mul(Z,A)) == False:
        return False
    return Z
```

Và hàm quan trọng nhất của chương trình là def Tim_Ma_Tran_Nghich_Dao(A): hàm có vai trò giúp tìm ra ma trận nghịch đảo thông qua giải thuật Gauss_Jodan. Ở cuối hàm sẽ so sánh xem ma trận mới tạo thành nhân với ma trận ban đầu có tạo thành ma trận đơn vị không. Nếu không bằng và trả về False, nếu bằng thì sẽ trả về ma trận nghịch đảo cần tìm.

```
 d = input("Nhap so dong: ")
c = input("Nhap so cot: ")
m = int(d)
n = int(c)
A = []
for i in range(0, m):
    A.append([])
    for j in range(0, n):
        A[i].append(0)
in_Put(A, m, n)
if m==n  and Tim_Ma_Tran_Nghich_Dao(A) is not False:
    print("Ma Tran Nghich dao cua A la: \n",Tim_Ma_Tran_Nghich_Dao(A))
else:
    print("Khong co ma tran nghich dao cua A")
```

Cuối cùng ở hàm main ta sẽ cho nhập vào ma trận cần tìm nghịch đảo với số dòng và số cột được nhập từ bàn phím. Kiểm tra xem nếu nó là ma trận vuông hay không và ma trận nhập vào có ma trận nghịch đảo hay không. Nếu có sẽ in ra ma trận nghịch đảo cần tìm. Nếu không sẽ in ra “Khong co ma tran nghich dao cua A”.

III. Cách chạy chương trình và kết quả chạy chương trình

1.Cách chạy chương trình:

```
Nhap so dong: 3
Nhap so cot: 3
```

Đầu tiên nhập vào số dòng và số cột từ bàn phím

```
A[ 0 ][ 0 ]:
1
A[ 0 ][ 1 ]:
2
A[ 0 ][ 2 ]:
3
A[ 1 ][ 0 ]:
5
A[ 1 ][ 1 ]:
2
A[ 1 ][ 2 ]:
1
A[ 2 ][ 0 ]:
3
A[ 2 ][ 1 ]:
6
A[ 2 ][ 2 ]:
1
```

Tiếp theo sẽ nhập vào giá trị của ma trận

```
Ma Tran Nghich dao cua A la:
 [[-0.0625, 0.25, -0.0625], [-0.03125, -0.125, 0.21875], [0.375, -0.0, -0.125]]
```

Cuối cùng in ra ma trận nghịch đảo cần tìm nếu có

2.Kết quả chạy chương trình:

```
Nhap so dong: 3
Nhap so cot: 3
A[ 0 ][ 0 ]:
1
A[ 0 ][ 1 ]:
2
A[ 0 ][ 2 ]:
3
A[ 1 ][ 0 ]:
5
A[ 1 ][ 1 ]:
2
A[ 1 ][ 2 ]:
1
A[ 2 ][ 0 ]:
3
A[ 2 ][ 1 ]:
6
A[ 2 ][ 2 ]:
1
Ma Tran Nghich dao cua A la:
 [[-0.0625, 0.25, -0.0625], [-0.03125, -0.125, 0.21875], [0.375, -0.0, -0.125]]
```