

ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN TOÁN ỨNG DỤNG VÀ TIN HỌC



BÁO CÁO CUỐI KỲ
KỸ THUẬT LẬP TRÌNH

CHỦ ĐỀ: XÂY DỰNG THƯ VIỆN
TIỆN ÍCH VỀ MA TRẬN

Giảng viên hướng dẫn: TS.Vũ Thành Nam

Sinh viên thực hiện: Lê Ngọc Hà

MSSV : 20216922

Mã lớp : 142297

Học kỳ : 2022.2

Hà Nội, 08/2023.

Lời cảm ơn

Lời đầu tiên, em xin đặc biệt bày tỏ lòng biết ơn sâu sắc đến thầy **Vũ Thành Nam**. Người đã trực tiếp giảng dạy, hướng dẫn em trong học phần “Kỹ thuật lập trình” này. Những bài học trên lớp cùng sự tận tâm, nhiệt huyết từ thầy đã giúp em tiếp thu tốt hơn môn học và biết cách áp dụng các kiến thức vào bài tập lớn cuối kỳ.

Em cũng muốn gửi lời cảm ơn đến Viện Toán ứng dụng và Tin học - Đại Học Bách Khoa Hà Nội đã tạo điều kiện cho em được học tập và phát triển bản thân trong một môi trường đầy năng động và thoả sức sáng tạo.

Cuối cùng, em xin gửi lời biết ơn tới những người bạn, các anh chị sinh viên Viện Toán ứng dụng và Tin học đã cho em rất nhiều lời khuyên bổ ích cùng những kinh nghiệm quý báu giúp em hạn chế được những sai sót không đáng có.

Do kiến thức và các kỹ năng của bản thân vẫn còn hạn hẹp nên báo cáo của em không tránh khỏi những thiếu sót và sai sót. Bởi vậy, em rất mong nhận được những ý kiến đóng góp quý báu từ thầy giáo để bài báo cáo được hoàn thiện hơn.

Em xin chân thành cảm ơn!

Sinh viên thực hiện

Lê Ngọc Hà

Mục lục

Lời cảm ơn	2
Tóm tắt chủ đề	4
1 Mô tả cấu trúc chương trình	5
1.1 Lớp Matrix.	5
1.2 Chương trình chính.	6
1.3 Các thao tác với ma trận.	7
2 Mô tả chi tiết	9
3 Các kỹ thuật lập trình đã vận dụng	19
4 Các tình huống kiểm thử	24
4.1 Chương trình chạy đúng.	24
4.2 Kiểm thử các lỗi chương trình.	33
5 Tổng kết	43
Tài liệu tham khảo	44

Tóm tắt chủ đề

Trong bài báo cáo cuối kỳ học phần “Kỹ thuật lập trình”, em đã chọn làm về chủ đề 1: “Xây dựng thư viện/tiện ích về ma trận.”

Em nhận thấy đây là một chủ đề rất hay và có tính áp dụng thực tế cao. Ma trận không còn xa lạ gì với bất kỳ học sinh, sinh viên nào, đặc biệt là các bạn theo khối ngành toán, công nghệ thông tin như em thì lại càng trở nên gần gũi và tiếp xúc nhiều hơn.

Với các kiến thức nền tảng về ma trận đã được học từ cấp 3 và trong môn học “Đại số tuyến tính” cùng với những kỹ năng lập trình được tích lũy từ học phần “Kỹ thuật lập trình”, em đã thực hành xây dựng một thư viện về ma trận.

Trong quá trình tìm hiểu và nghiên cứu về các thuật toán liên quan đến ma trận như: phép cộng, trừ, nhân ma trận, tính định thức, ma trận nghịch đảo, chuyển vị... em đã thiết kế được một thư viện tích hợp các phương pháp tính toán ma trận cần thiết. Ngoài ra, em cũng quan tâm đến việc tối ưu hoá cho thư viện để tăng tốc độ, giảm bớt chi phí dữ liệu, giúp cho việc sử dụng thư viện dễ dàng và tiện lợi hơn.

Bài báo cáo này là một cơ hội tốt để em áp dụng lý thuyết vào thực hành, để hiểu rõ các kiến thức đã được dạy và biết cách làm thế nào để thiết kế một chương trình. Báo cáo gồm có 5 phần chính, tất cả các mã nguồn được thực hiện trên ngôn ngữ lập trình Python. Em hy vọng thư viện do mình thiết kế sẽ đem đến kết quả chính xác nhất và giúp cho việc xử lý dữ liệu ma trận không còn gặp nhiều khó khăn nữa.

1 Mô tả cấu trúc chương trình

Chương trình được viết bằng ngôn ngữ lập trình Python, định nghĩa lớp Matrix để thực hiện các thao tác trên ma trận. Dưới đây là cấu trúc và mô tả chương trình cho chủ đề "Xây dựng thư viện/tiện ích về ma trận":

1.1 Lớp Matrix.

Thuộc tính:

- rows: Số hàng của ma trận.
- cols: Số cột của ma trận.
- data: Dữ liệu của ma trận được lưu dưới dạng danh sách 2 chiều.

Các phương thức:

- `__init__(self, rows, cols)`: Khởi tạo một đối tượng matrix với số hàng (rows) và số cột (cols).
- `__str__(self)`: Trả về chuỗi biểu diễn ma trận.
- `input_from_user(self)`: Nhập ma trận từ bàn phím.
- `input_from_file(self, filename)`: Nhập ma trận từ tệp tin.
- `output_to_file(self, output_filename, output)`: Các kết quả được lưu vào một tệp tin.
- `addition(self, other)`: Tính tổng hai ma trận.
- `subtraction(self, other)`: Tính hiệu hai ma trận.
- `multiply(self, other)`: Tính tích hai ma trận.

- `multiply_by_number(self, number: float)`: Nhân ma trận với một số.
- `gauss(self)`: Phương pháp Gauss để phục vụ việc tính định thức ma trận và tìm hạng ma trận.
- `calculate_determinant(self)`: Tính định thức ma trận.
- `calculate_rank(self)`: Tính hạng ma trận.
- `calculate_inverse(self)`: Tính ma trận nghịch đảo.
- `calculate_transpose(self)`: Tính ma trận chuyển vị.
- `from_list(data: list)`: Phương thức tĩnh dùng để tạo một đối tượng ma trận từ một danh sách dữ liệu đã cho phục vụ cho việc tính ma trận nghịch đảo.

1.2 Chương trình chính.

Bước 1: Đầu tiên, người dùng sẽ được yêu cầu nhập ma trận A từ bàn phím hoặc từ một tệp tin. Ma trận A được hiển thị ra màn hình và ghi vào tệp tin đầu ra (nếu có).

Bước 2: Chương trình hiển thị menu với các lựa chọn để thực hiện các phép tính toán trên ma trận hoặc thoát khỏi chương trình:

- 1: Cộng hai ma trận.
- 2: Hiệu hai ma trận.
- 3: Nhân hai ma trận.
- 4: Nhân một số với ma trận.
- 5: Tính định thức của ma trận.
- 6: Tính hạng của ma trận.
- 7: Tính ma trận nghịch đảo.
- 8: Tính ma trận chuyển vị.
- 9: Thoát khỏi chương trình.

Bước 3: Tùy thuộc vào lựa chọn của người dùng, chương trình sẽ thực hiện phép tính tương ứng.

- Nếu lựa chọn (1), (2), (3) sẽ được yêu cầu nhập ma trận B từ bàn phím hoặc từ một tệp tin. Ma trận B được hiển thị ra màn hình và ghi vào tệp tin đầu ra (nếu có).
- Nếu lựa chọn (4) sẽ được yêu cầu nhập số nguyên hoặc số thực muốn nhân với ma trận A.
- Nếu lựa chọn (5), (6), (7), (8) chương trình thực hiện các yêu cầu tương ứng, người dùng không cần thao tác gì thêm.
- Nếu lựa chọn (9) khi người dùng muốn chuyển sang các phép tính toán với dữ liệu khác. Kết thúc chương trình hiện tại và quay lại nhập ma trận mới.
- Nếu người dùng không chọn đúng các lựa chọn từ (1) đến (9), chương trình sẽ hiển thị "Lựa chọn không hợp lệ" và quay lại menu yêu cầu nhập lại.

Bước 4: Sau mỗi phép tính, kết quả được hiển thị trên màn hình và ghi vào tệp tin chương trình đã mặc định (*output.txt*).

1.3 Các thao tác với ma trận.

- Để nhập ma trận từ bàn phím, người dùng cần cung cấp số hàng và số cột của ma trận sau đó nhập các giá trị cho từng phần tử.
- Để nhập ma trận từ tệp tin, người dùng cần chuẩn bị tệp tin đầu vào với định dạng đúng. Dòng đầu tiên của tệp tin chứa hai giá trị nguyên dương là số hàng và số cột của ma trận, các dòng tiếp theo chứa giá trị của ma trận. Các giá trị cách nhau bởi khoảng trắng.
- Nếu ma trận không đúng định dạng, chương trình sẽ báo lỗi cụ thể trên màn hình để người dùng nắm rõ và sửa đổi cho đúng.

- Đối với mỗi lựa chọn, chương trình sẽ tạo bản sao của ma trận A ban đầu. Điều này đảm bảo rằng bất kỳ thay đổi nào trên bản sao sẽ không ảnh hưởng đến ma trận A ban đầu và giữ cho ma trận A nguyên vẹn sau mỗi phép toán.

2 Mô tả chi tiết

1. Phương thức init.

```
1 def __init__(self, rows, cols):
2     self.rows = rows
3     self.cols = cols
4     self.data = [[0] * cols for _ in range(rows)]
```

- Đây là phương thức khởi tạo cho lớp Matrix.
- Nhận hai tham số là rows và cols, tương ứng là số hàng và số cột của ma trận.
- Khởi tạo các thuộc tính self.rows và self.cols để lưu trữ số lượng hàng và cột, đồng thời tạo thuộc tính self.data dưới dạng danh sách hai chiều.

2. Phương thức str.

```
1 def __str__(self):
2     output = "\n"
3     for row in self.data:
4         for element in row:
5             output += f'{format(round(element, 9), "> 15")}'
6         output += '\n'
7     return output
```

- Phương thức này dùng để biểu diễn chuỗi của ma trận sau khi được chuyển đổi.
- Sử dụng vòng lặp for để lặp qua danh sách hai chiều (self.data).
- Các phần tử sẽ được hiển thị với chiều rộng 15 ký tự và lấy 9 chữ số thập phân sau dấu phẩy.
- Các phần tử định dạng xong sẽ được thêm vào output, thêm ký tự xuống dòng ở cuối mỗi hàng để phân tách các hàng trong chuỗi.

3. Phương thức `input_from_user`.

```

1 def input_from_user(self):
2     while True:
3         try:
4             if self.rows <= 0 or self.cols <= 0:
5                 print("Số hàng và số cột phải là số nguyên dương. Vui
6                 long nhập lại.")
7             else:
8                 break
9         except ValueError:
10            print("Giá trị không hợp lệ: các giá trị phải là số nguyên
11            dương. Vui long nhập lại.")
12
13    for i in range(self.rows):
14        row = []
15        for j in range(self.cols):
16            while True:
17                try:
18                    value = float(input(f"Nhap gia tri [{i + 1}][{j +
19                    1}]: "))
20                except ValueError:
21                    print("Giá trị không hợp lệ: các giá trị phải là số
22                    nguyên hoặc số thực. Vui long nhập lại.")
23                row.append(value)
24            self.data[i] = row

```

Đây là phương thức yêu cầu người dùng nhập ma trận từ bàn phím.

Bước 1: Bắt đầu bằng vòng lặp vô hạn (`while True`) để yêu cầu nhập số hàng và số cột.

Bước 2:

- Nếu số hàng hoặc số cột nhỏ hơn hoặc bằng 0, người dùng sẽ nhận được thông báo lỗi và yêu cầu nhập lại.
- Nếu số hàng và số cột hợp lệ, vòng lặp sẽ `break` và tới bước tiếp theo.

Bước 3: Chương trình tiếp tục với vòng lặp `for` để lặp qua từng hàng, từng cột trong mảng hai chiều.

Bước 4: Chương trình sẽ yêu cầu nhập giá trị cho từng phần tử của mảng.

- Nếu giá trị nhập vào không có dạng `int` hoặc `float`, chương trình sẽ báo lỗi cho người dùng và yêu cầu nhập lại.
- Nếu giá trị hợp lệ, thêm giá trị đó vào hàng hiện tại và tiếp tục vòng lặp để nhập giá trị cho cột tiếp theo.

Bước 5: Sau khi hoàn thành vòng lặp con, hàng hiện tại được thêm vào mảng hai chiều chính (`self.data[i] = row`).

4. Phương thức `input_from_file`.

```

1 def input_from_file(self, filename):
2     try:
3         with open(filename) as f:
4             count = 0
5
6             for line in f:
7                 if count == 0:
8                     size = line.strip().split()
9                     if len(size) != 2 or not size[0].isdigit() or not
10 size[1].isdigit():
11                         raise ValueError("Dong dau tien cua file chi
12 chua hai gia tri nguyen duong (so hang, so cot) va cach nhau boi
13 khoang trang.")
14                     self.rows = int(size[0])
15                     self.cols = int(size[1])
16                     self.data = [[0] * self.cols for _ in range(self.
17 rows)]
18                 else:
19                     values = line.strip().split()
20                     if len(values) != self.cols:
21                         raise ValueError("So luong gia tri khong tuong
22 ung voi so cot cua ma tran.")
23                     try:
24                         row = [float(value) for value in values]
25                     except ValueError:
26                         raise ValueError("Khong the chuyen doi gia tri
27 sang so thuc.")
28                     self.data[count - 1] = row
29                     count += 1
30
31             if count - 1 != self.rows:
32                 raise ValueError("So luong dong du lieu trong file khong
33 tuong ung voi so hang cua ma tran.")
34
35     except FileNotFoundError:
36         raise FileNotFoundError("Tep tin khong ton tai, vui longkiem
37 tra lai.")

```

Đây là phương thức yêu cầu người dùng nhập ma trận từ tệp tin đã chuẩn bị.

Bước 1: Chương trình sẽ mở tệp tin, đoạn mã đọc từng dòng trong tệp từ dòng đầu tiên đến các dòng tiếp theo.

Bước 2: Sử dụng vòng lặp for đầu tiên với `count = 0` để lấy số hàng và số cột.

- Nếu giá trị không hợp lệ, chương trình thông báo lỗi cụ thể.

- Nếu giá trị hợp lệ, khởi tạo mảng hai chiều (self.data), lặp qua các dòng còn lại.

Bước 3: Với mỗi dòng tiếp theo, đoạn mã sẽ tách các giá trị. Nếu số lượng giá trị không tương ứng với số cột của ma trận, chương trình sẽ báo lỗi.

Bước 4: Đoạn mã chuyển đổi các giá trị tách ra từ dạng chuỗi sang số thực.

- Nếu chuyển đổi không thành công, chương trình sẽ báo lỗi.

- Nếu chuyển đổi thành công, dòng đó sẽ được thêm vào mảng tại vị trí tương ứng với số hàng hiện tại (count - 1).

Bước 5: Sau khi đọc hết tệp, đoạn mã kiểm tra xem số lượng dòng đã đọc có bằng với số hàng đã xác định không.

- Nếu không tương ứng sẽ thông báo ValueError.

- Nếu tệp không tồn tại, chương trình hiển thị lỗi tới người dùng.

5. Phương thức output_to_file.

```
1 def output_to_file(self, output_filename, output):
2     with open(output_filename, "a", encoding="utf-8") as f:
3         lines = output.split('\n')
4         for line in lines:
5             f.write(line + '\n')
6     print("Ket qua duoc luu vao file:", output_filename)
```

Phương thức này dùng để lưu kết quả từ lựa chọn của người dùng vào tệp tin.

Bước 1: Chương trình sẽ mở file output với chế độ "a" (append mode) để thêm dữ liệu vào cuối tệp. Dữ liệu sẽ được ghi vào tệp mà không xóa nội dung trước đó.

Bước 2: Nối thêm output chuỗi vào tệp đã chỉ định bằng cách sử dụng mã hóa (UTF-8).

Bước 3: Sử dụng vòng lặp for lặp qua từng dòng trong lines, trong mỗi lần lặp các dòng sẽ được ghi vào tệp f và cách nhau bởi ký tự xuống dòng.

Bước 4: Sau khi vòng lặp hoàn tất, chương trình sẽ in ra thông báo kết quả đã được lưu vào tệp.

6. Phương thức addition, subtraction, multiply.

```

1 def addition(self, other):
2     if self.rows != other.rows or self.cols != other.cols:
3         raise ValueError("Hai ma tran khong cung kich thuoc.")
4     result = Matrix(self.rows, self.cols)
5     for i in range(self.rows):
6         for j in range(self.cols):
7             result.data[i][j] = self.data[i][j] + other.data[i][j]
8     return result

```

```

1 def subtraction(self, other):
2     if self.rows != other.rows or self.cols != other.cols:
3         raise ValueError("Hai ma tran khong cung kich thuoc.")
4     result = Matrix(self.rows, self.cols)
5     for i in range(self.rows):
6         for j in range(self.cols):
7             result.data[i][j] = self.data[i][j] - other.data[i][j]
8     return result

```

```

1 def multiply(self, other):
2     if self.cols != other.rows:
3         raise ValueError("So cot cua ma tran A phai bang so hang cua ma
4         tran B.")
5     result = Matrix(self.rows, other.cols)
6     for i in range(self.rows):
7         for j in range(other.cols):
8             total = 0
9             for k in range(self.cols):
10                 total += self.data[i][k] * other.data[k][j]
11             result.data[i][j] = total
12     return result

```

Các phương pháp này thực hiện các phép toán số học cơ bản giữa hai ma trận.

Bước 1: Kiểm tra kích thước hai ma trận có tương ứng với nhau không, bằng cách so sánh số hàng và số cột.

- Nếu không bằng nhau chương trình sẽ báo lỗi và không tiếp tục chạy nữa.
- Nếu hai ma trận đã cùng kích cỡ, khởi tạo ma trận (result).

Bước 2: Sử dụng vòng lặp for để duyệt qua từng phần tử có cùng vị trí.

Bước 3: Tại mỗi vị trí (i, j), giá trị của phần tử trong ma trận (result) sẽ được tính bằng tổng/hiệu/tích của hai phần tử tại cùng vị trí trong hai ma trận.

Bước 4: Kết quả cuối cùng trả về là ma trận (result).

7. Phương thức multiply_by_number.

```

1 def multiply_by_number(self, number: float):
2     if not isinstance(number, float):
3         raise ValueError("Số nhập vào không hợp lệ. Vui lòng nhập một số
         nguyên hoặc số thực.")
4     for i in range(self.rows):
5         for j in range(self.cols):
6             self.data[i][j] *= number
7     return self

```

Phương thức này dùng để nhân ma trận với một số vô hướng.

Bước 1: Đầu tiên, đoạn mã kiểm tra số nhập vào.

- Nếu không phải số thực, báo ValueError và yêu cầu nhập lại.
- Nếu số nhập vào hợp lệ, chạy vòng for để duyệt qua từng phần tử trong ma trận.

Bước 2: Tại mỗi vị trí (i, j), giá trị của phần tử trong ma trận (self) sẽ được nhân với (number).

Bước 3: Kết quả cuối cùng trả về là ma trận (self).

8. Phương pháp Gauss.

```

1 def gauss(self):
2     if (self.rows == 0) or (self.cols == 0) or (len(self.data) != self.
         rows) or (any(len(row) != self.cols for row in self.data)):
3         raise ValueError("Kích thước ma trận không tương thích với dữ
         liệu.")
4
5     m = self.rows
6     n = self.cols
7     rank = min(m, n)
8     A = self.data
9     eps = 1e-15
10
11     # Tìm phần tử có giá trị tuyệt đối lớn nhất
12     for i in range(rank):
13         max_row = i
14         for j in range(i + 1, m):
15             if abs(A[j][i]) > abs(A[max_row][i]):
16                 max_row = j
17     # Kiểm tra phần tử đường chéo chính
18     if abs(A[i][i]) < eps:
19         for k in range(i + 1, m):
20             # Kiểm tra phần tử dưới đường chéo chính
21             if abs(A[k][i]) > eps:
22                 A[i], A[k] = A[k], A[i]
23                 break

```

```

24         else:
25             rank -= 1
26             continue
27
28     # Chuẩn hóa các hàng
29     for j in range(i + 1, m):
30         factor = A[j][i] / A[i][i]
31         for k in range(i + 1, n):
32             A[j][k] -= factor * A[i][k]
33
34     return rank

```

Sử dụng kết quả của phương pháp Gauss để tính định thức và hạng của ma trận.

Bước 1: Chương trình kiểm tra kích thước ma trận và đảm bảo dữ liệu đủ để tính toán. Nếu ma trận không hợp lệ sẽ thông báo ValueError.

Bước 2: Xác định số hàng (m), số cột (n), số cột tối thiểu cần duyệt để tính hạng ma trận (rank), khởi tạo ma trận (A), độ sai số (eps).

Bước 3: Duyệt qua từng hàng i trong ma trận. Tại mỗi bước, tìm phần tử có giá trị tuyệt đối lớn nhất trong cột.

- Nếu phần tử đường chéo chính $A[i][i]$ gần bằng 0, đổi hàng hiện tại với hàng có giá trị không gần bằng 0 khác.

- Nếu không tìm được, giảm (rank) đi 1 và tiếp tục với hàng tiếp theo.

Bước 4: Chuẩn hóa các hàng phía dưới hàng i để giảm phần tử dưới đường chéo chính $A[j][i]$ về 0, đưa ma trận về dạng tam giác trên.

Bước 5: Kết quả cuối cùng trả về (rank) của ma trận.

9. Phương thức calculate_determinant.

```

1 def calculate_determinant(self):
2     if self.rows != self.cols:
3         raise ValueError("Ma tran khong phai ma tran vuong")
4     if self.rows == 0:
5         return "Ma tran rong"
6     n = self.rows
7     det = 1
8     rank = self.gauss()
9     if rank < n:
10        return "Dinh thuc bang 0"
11    for i in range(rank):
12        det *= self.data[i][i]
13    return det

```

Đây là phương thức dùng để tính định thức của ma trận.

Bước 1: So sánh số hàng và số cột. Nếu không phải ma trận vuông, chương trình thông báo ValueError.

Bước 2: Kiểm tra số hàng. Nếu ma trận rỗng, trả về thông báo.

Bước 3: Khởi tạo biến (det), khởi tạo biến (rank) bằng cách gọi phương thức gauss() từ lớp hiện tại.

Bước 4: Nếu (rank) nhỏ hơn số hàng/số cột thì định thức bằng 0 vì không thể tạo nên đủ các hàng độc lập tuyến tính.

Bước 5: Nếu (rank) bằng số hàng/số cột, sử dụng vòng for nhân các phần tử trên đường chéo chính của ma trận sau khi được chuẩn hóa.

Bước 6: Kết quả cuối cùng trả về là (det).

10. Phương thức calculate_rank.

```

1 def calculate_rank(self):
2     if self.rows == 0 or self.cols == 0:
3         return 0
4     return self.gauss()

```

Đây là phương thức dùng để tính hạng của ma trận.

Bước 1: Kiểm tra nếu số hàng hoặc số cột bằng 0, phương thức trả về 0 vì ma trận không có phần tử nào và bậc của ma trận bằng 0.

Bước 2: Nếu ma trận không rỗng, chương trình gọi phương thức gauss() từ lớp hiện tại để tính bậc của ma trận.

Bước 3: Kết quả cuối cùng trả về là (rank) của ma trận.

11. Phương thức calculate_inverse.

```

1 @staticmethod
2 def from_list(data: list):
3     m = Matrix(len(data), len(data[0]))
4     m.data = data
5     return m

```



```

1 # Hàm tính ma trận nghịch đảo
2 def calculate_inverse(self):
3     if self.rows != self.cols:
4         raise ValueError("Ma trận không phải ma trận vuông")
5
6     # Tạo ma trận đơn vị
7     identity = [[0 for j in range(self.cols)] for i in range(self.rows)]
8     for i in range(self.rows):
9         identity[i][i] = 1
10
11     A = self.data
12     B = identity
13     n = self.rows
14
15     for i in range(n):
16         # Kiểm tra phần tử dương chéo chính
17         if A[i][i] == 0:
18             for k in range(i + 1, n):
19                 if A[k][i] != 0:
20                     A[i], A[k] = A[k], A[i]
21                     B[i], B[k] = B[k], B[i]
22                     break
23             else:
24                 raise ValueError("Ma trận không thể nghịch đảo")
25
26     pivot = A[i][i]
27
28     # Chuẩn hóa hàng i để phần tử dương chéo chính bằng 1
29     for j in range(n):
30         A[i][j] /= pivot
31         B[i][j] /= pivot
32
33     # Dưa các phần tử dưới dương chéo chính về 0
34     for k in range(n):
35         if k == i:
36             continue
37         factor = A[k][i]
38         for j in range(n):
39             A[k][j] -= factor * A[i][j]
40             B[k][j] -= factor * B[i][j]
41
42     return Matrix.from_list(B)

```

Các phương thức trên phục vụ cho việc tính ma trận nghịch đảo.

Bước 1: Kiểm tra số hàng khác số cột thì thông báo ValueError.

Bước 2: Tạo ma trận đơn vị (identity), các phần tử trên đường chéo chính bằng 1 và các phần tử còn lại bằng 0.

Bước 3: Khởi tạo hai ma trận (A) và (B).

Bước 4: Tìm ma trận nghịch đảo bằng phương pháp Gauss-Jordan:

- Duyệt qua từng hàng i trong ma trận.

- Nếu phần tử $A[i][i]$ bằng 0, tìm hàng khác 0 trong cùng cột và đổi chỗ hàng.
- Nếu không tìm được, hiển thị ValueError.
- Chuẩn hóa các phần tử dưới đường chéo chính bằng cách chia hàng i cho giá trị của phần tử đường chéo chính, đồng thời thực hiện cùng thao tác trên ma trận B để duy trì tính nghịch đảo.
- Tiến hành loại bỏ các phần tử dưới đường chéo chính bằng cách trừ bỏ bội số của hàng i từ các hàng khác, đồng thời thực hiện cùng thao tác trên ma trận B.

Bước 5: Sau khi duyệt qua tất cả các hàng, ma trận A sẽ trở thành ma trận đơn vị và ma trận B sẽ là ma trận nghịch đảo của ma trận ban đầu.

Bước 6: Kết quả trả về ma trận nghịch đảo dưới dạng một đối tượng Matrix bằng cách sử dụng phương thức `from_list(B)`.

12. Phương thức `calculate_transpose`.

```

1 def calculate_transpose(self):
2     result = Matrix(self.cols, self.rows)
3     for i in range(self.rows):
4         for j in range(self.cols):
5             result.data[j][i] = self.data[i][j]
6     return result

```

Đây là phương thức dùng để tìm ma trận chuyển vị.

Bước 1: Tạo một đối tượng matrix mới có số hàng bằng số cột và số cột bằng số hàng của ma trận ban đầu.

Bước 2: Sử dụng hai vòng lặp for lồng nhau để duyệt qua từng phần tử của ma trận hiện tại theo hàng và cột.

Bước 3: Gán giá trị của phần tử (`self.data[i][j]`) vào phần tử tương ứng của ma trận chuyển vị (`result.data[j][i]`).

Bước 4: Kết quả trả về là ma trận chuyển vị của ma trận ban đầu.

3

Các kỹ thuật lập trình đã vận dụng

Sau quá trình kết hợp với slide bài giảng và thực hiện các thao tác lập trình, em đã vận dụng các kỹ thuật lập trình sau để đem đến một "chương trình tốt" nhất có thể.

3.1 Lập trình tốt.

Đề tài "Xây dựng thư viện/tiện ích về ma trận" được xây dựng dựa trên một chương trình đáp ứng đủ 4 yếu tố để có thể lập trình tốt:

- **Tính chính xác:** Chương trình sau mỗi lần thực hiện các phép tính toán đều mang lại kết quả đúng, có khả năng xử lý các ma trận có kích thước lớn hoặc ma trận có giá trị rất gần với 0.
- **Tính ổn định:** Chương trình luôn xác thực các dữ liệu đầu vào, kiểm tra tính hợp lệ của dữ liệu trước khi thực hiện các phép tính.
- **Khả năng nâng cấp:** Ngoài các phép toán cơ bản trên ma trận (cộng, trừ, nhân...) chương trình còn có các chức năng mới (ma trận nghịch đảo, tính định thức ma trận...)
- **Tái sử dụng:** Chương trình hiện tại đã được thiết kế thành một lớp Matrix, điều này tạo ra tính tái sử dụng trong mã nguồn.

3.2 Hình mẫu lập trình.

- Chương trình xây dựng trên nguyên lý lập trình hướng đối tượng bằng Python.
- Tạo một lớp Matrix, trong đó định nghĩa các phương thức hoạt động trên ma trận. Các phương thức này tương tác với nhau để thực hiện các chức năng của chương trình.
- Một số ưu điểm của lập trình hướng đối tượng: tính tái sử dụng, quản lý code dễ dàng, tính trừu tượng, đóng gói, kế thừa, đa hình...

3.3 Nguyên lý lập trình cấu trúc.

Chương trình đã vận dụng các nguyên lý lập trình sau:

- Nguyên lý lệnh có cấu trúc.
- Nguyên lý tối thiểu.
- Nguyên lý nhất quán.
- Nguyên lý an toàn.

3.4 Phong cách lập trình.

1. Nguyên tắc chung.

Chương trình đáp ứng đủ các nguyên tắc chung: đơn giản, trực tiếp, rõ ràng và cấu trúc khá tốt.

2. Một số quy tắc khác.

- Tính nhất quán: Tuân thủ quy tắc đặt tên trong toàn bộ chương trình.
- Tính khúc triết: Mỗi phương thức đều có nhiệm vụ rõ ràng.
- Tính rõ ràng: Các chú thích đều dễ hiểu.
- Tính bao đóng.

3. Cấu trúc mã nguồn.

Một mã nguồn tốt là một mã nguồn dễ đọc, chương trình đã sử dụng các quy tắc sau đây để cấu trúc mã nguồn hoàn thiện nhất:

- **Khoảng trắng (Spacing):** Chương trình sử dụng các khoảng trắng phù hợp và nhất quán trong suốt quá trình thực hiện.

- **Cách lề (Indentation):** Cách lề hợp lý để tránh nhầm lẫn giữa các cấu trúc.

VD: Cặp lệnh (try, except) và (if, else) được cách lề thẳng nhau, góp phần làm cho đoạn mã dễ nhìn hơn.

```

1 def input_from_user(self):
2     while True:
3         try:
4             if self.rows <= 0 or self.cols <= 0:
5                 print("...")
6                 return None
7             else:
8                 break
9         except ValueError:
10            print("...")

```

- **Cách đoạn (Paragraph):** Giữa các phương thức đều cách nhau 2 dòng để chương trình nhìn thoáng và dễ tìm kiếm khi muốn thay đổi hoặc sửa lỗi.

VD: Cách đoạn giữa hai phương thức init và phương thức str.

```

1 # 1. Ham khoi tao lop Matrix
2 def __init__(self, rows, cols):
3     self.rows = rows
4     self.cols = cols
5     self.data = [[0] * cols for _ in range(rows)]
6
7
8 # 2. Ham hien thi ma tran duoi dang chuoi
9 def __str__(self):
10    output = "\n"
11    ...

```

- **Biểu thức (Expressions):** Dùng các biểu thức dạng nguyên bản, dùng () để tránh nhầm lẫn.

VD: Dùng dấu () để các biểu thức dễ nhìn hơn.

```

1 if (self.rows != other.rows) or (self.cols != other.cols):
2     raise ValueError("Hai ma tran khong cung kich thuoc.")

```

4. Đặt tên và chú thích.

- Cách đặt tên (Name):

- Dùng tên gọi nhớ, có tính miêu tả cho các phương thức.
VD: addition, multiply, init, str...
- Dùng tên nhất quán cho các biến cục bộ.
VD: Dùng các chữ cái i, j, k cho các biến chạy vòng lặp.
- Dùng chữ hoa, chữ thường nhất quán.
- Dùng phong cách nhất quán khi ghép từ.
VD: calculate_inverse, multiply_by_number, matrix_a...
- Dùng động từ cho tên hàm.
VD: input_from_file, input_from_user, output_to_file...

- Chú thích (Comments):

- Ở đầu các phương thức đều có chú thích tổng thể. Các chú thích trong chương trình dùng để thể hiện phương thức đang làm gì chứ không phải làm như nào.
VD:

```
1 # Ham tinh tong hai ma tran
2 def addition(self, other):
3
4 # Ham tinh hieu hai ma tran
5 def subtraction(self, other):
```

- Đối với những phương thức cần tính toán phức tạp và có nhiều dòng code, chương trình sẽ thêm chú thích từng dòng phù hợp để đoạn mã dễ hiểu hơn.
VD:

```
1 # Phuong phap Gauss
2 def gauss(self):
3     ...
4     # Tim phan tu co gia tri tuyet doi lon nhat
5     for i in range(rank):
6         ...
7
```

```
8      # Kiem tra phan tu tren duong cheo chinh
9      if abs(A[i][i]) < eps:
10         ...
11         # Kiem tra phan tu duoi duong cheo chinh
12         if abs(A[k][i]) > eps:
13             ...
14     # Chuan hoa cac hang
15     for j in range(i + 1, m):
```

- Viết chú thích tương ứng với code, thay đổi khi bản thân code thay đổi.

3.5 Kiểm thử, tối ưu mã và tinh chỉnh.

1. Kỹ thuật kiểm thử.

- Xử lý ngoại lệ: Sử dụng cấu trúc (try, except) để xử lý các trường hợp ngoại lệ khi nhập giá trị từ người dùng hoặc từ tệp tin.
- Các ngoại lệ như ValueError và FileNotFoundError được xử lý thông qua các khối except tương ứng.
- Kiểm tra tính hợp lệ của đầu vào ứng với mỗi phương thức trước khi thực hiện phép toán.

2. Kỹ thuật tối ưu mã và tinh chỉnh.

- copy.deepcopy(): Vận dụng thư viện copy để sao chép đối tượng ma trận trước khi thực hiện các phép toán, tránh làm thay đổi trực tiếp ma trận ban đầu.
- @staticmethod: Phương thức tĩnh tạo ma trận từ danh sách dữ liệu mà không cần khởi tạo đối tượng.
- Menu: Cho phép người dùng dễ dàng chọn các tùy chọn và tương tác với chương trình một cách rõ ràng, đồng thời giúp người lập trình dễ dàng quản lý và mở rộng.
- Phương pháp Gauss: Tận dụng tính chất của ma trận bậc thang để giảm bớt số lượng phép toán cần thực hiện, làm tăng hiệu suất tính toán.
- Tối ưu các biểu thức điều kiện...

4

Các tình huống kiểm thử

4.1 Chương trình chạy đúng.

1. Nhập ma trận từ bàn phím.

- Nhập ma trận A từ bàn phím:

```
Nhập tên tệp tin (filename.txt) cho ma trận A hoặc nhấn Enter để nhập từ bàn phím:
Nhập số hàng của ma trận A: 3
Nhập số cột của ma trận A: 3
Nhập giá trị [1][1]: 1
Nhập giá trị [1][2]: 4
Nhập giá trị [1][3]: 7
Nhập giá trị [2][1]: 8
Nhập giá trị [2][2]: 9
Nhập giá trị [2][3]: 6
Nhập giá trị [3][1]: 5
Nhập giá trị [3][2]: 2
Nhập giá trị [3][3]: 1
Ma trận A là:
```

1.0	4.0	7.0
8.0	9.0	6.0
5.0	2.0	1.0

- Nhập ma trận B từ bàn phím:

```
Nhập tên tệp tin (filename.txt) cho ma trận B hoặc nhấn Enter để nhập từ bàn phím:
Nhập số hàng của ma trận B: 3
Nhập số cột của ma trận B: 3
Nhập giá trị [1][1]: 2
Nhập giá trị [1][2]: 5
Nhập giá trị [1][3]: 8
Nhập giá trị [2][1]: 9
Nhập giá trị [2][2]: -3
Nhập giá trị [2][3]: -2
Nhập giá trị [3][1]: 7
Nhập giá trị [3][2]: -9
Nhập giá trị [3][3]: 0
Ma trận B là:
```

2.0	5.0	8.0
9.0	-3.0	-2.0
7.0	-9.0	0.0

- Tổng hai ma trận A và B:

Tổng hai ma trận là:

		3.0	9.0	15.0
		17.0	6.0	4.0
		12.0	-7.0	1.0

- Hiệu hai ma trận A và B:

Hiệu hai ma trận là:

		-1.0	-1.0	-1.0
		-1.0	12.0	8.0
		-2.0	11.0	1.0

- Tích hai ma trận A và B:

Tích hai ma trận là:

		87.0	-70.0	0.0
		139.0	-41.0	46.0
		35.0	10.0	36.0

- Tích ma trận A với một số:

Vui lòng chọn: 4
 Nhập số muốn nhân với ma trận: 20
 Tích của ma trận với một số là:

		20.0	80.0	140.0
		160.0	180.0	120.0
		100.0	40.0	20.0

Kết quả đã được lưu vào file: output.txt

- Kết quả định thức, hạng ma trận A, ma trận nghịch đảo, ma trận chuyển vị của ma trận A:

Định thức của ma trận đã nhập là: -118.0

Hạng của ma trận đã nhập là: 3

Ma trận nghịch đảo của ma trận đã nhập là:

0.025423729	-0.084745763	0.330508475
-0.186440678	0.288135593	-0.423728814
0.245762712	-0.152542373	0.194915254

Ma trận chuyển vị của ma trận đã nhập là:

1.0	8.0	5.0
4.0	9.0	2.0
7.0	6.0	1.0

2. Nhập ma trận từ tệp tin.

Ma trận cỡ 6x6:

- Nhập ma trận A từ tệp tin (*input_a.txt*):

Nhập tên tệp tin (filename.txt) cho ma trận A hoặc nhấn Enter để nhập từ bàn phím: input_a.txt
Ma trận A là:

1.0	1.0	2.0	5.0	8.0	7.0
3.0	5.0	6.0	5.0	9.0	0.0
7.0	2.0	3.0	8.0	7.0	9.0
7.0	8.0	9.0	2.0	2.0	3.0
2.0	3.0	8.0	7.0	4.0	1.0
8.0	2.0	3.0	1.0	4.0	7.0

- Nhập ma trận B từ tệp tin (*input_b.txt*):

```

Vui lòng nhập:
Nhập tên tệp tin (filename.txt) cho ma trận B hoặc nhấn Enter để nhập từ bàn phím: input_b.txt
Ma trận B là:

      1.0      4.0      7.0      5.0      3.0      2.0
      2.0      8.0      6.0      7.0      9.0      3.0
      5.0      3.0      1.0      7.0      8.0      9.0
      1.0      4.0      2.0      3.0      8.0      6.0
      1.0      8.0      6.0      4.0      7.0      3.0
      9.0      6.0      4.0      3.0      7.0      8.0

```

- Tổng hai ma trận A và B:

```

Tổng hai ma trận là:

      2.0      5.0      9.0      10.0      11.0      9.0
      5.0     13.0     12.0     12.0     18.0      3.0
     12.0      5.0      4.0     15.0     15.0     18.0
      8.0     12.0     11.0      5.0     10.0      9.0
      3.0     11.0     14.0     11.0     11.0      4.0
     17.0      8.0      7.0      4.0     11.0     15.0

```

- Hiệu hai ma trận A và B:

```

Hiệu hai ma trận là:

      0.0     -3.0     -5.0      0.0      5.0      5.0
      1.0     -3.0      0.0     -2.0      0.0     -3.0
      2.0     -1.0      2.0      1.0     -1.0      0.0
      6.0      4.0      7.0     -1.0     -6.0     -3.0
      1.0     -5.0      2.0      3.0     -3.0     -2.0
     -1.0     -4.0     -1.0     -2.0     -3.0     -1.0

```

- Tích hai ma trận A và B:

```

Tích hai ma trận là:

      89.0     144.0     101.0      94.0     173.0     133.0
      57.0     162.0     121.0     143.0     205.0     132.0
     122.0     195.0     158.0     149.0     239.0     188.0
      99.0     161.0     134.0     177.0     216.0     161.0
      68.0     122.0      82.0     127.0     188.0     147.0
      95.0     135.0     125.0     115.0     151.0     123.0

```

- Tích ma trận A với một số:

```
Vui lòng chọn: 4
Nhập số muốn nhân với ma trận: 20
Tích của ma trận với một số là:
```

20.0	20.0	40.0	100.0	160.0	140.0
60.0	100.0	120.0	100.0	180.0	0.0
140.0	40.0	60.0	160.0	140.0	180.0
140.0	160.0	180.0	40.0	40.0	60.0
40.0	60.0	160.0	140.0	80.0	20.0
160.0	40.0	60.0	20.0	80.0	140.0

Kết quả đã được lưu vào file: output.txt

- Kết quả định thức, hạng ma trận A, ma trận nghịch đảo, ma trận chuyển vị của ma trận A:

```
Định thức của ma trận đã nhập là: 67868.99999999997
Hạng của ma trận đã nhập là: 6
Ma trận nghịch đảo của ma trận đã nhập là:
```

-0.183309022	0.068691155	0.075586792	-0.067011449	-0.001812315	0.115104098
0.034537123	0.037616585	0.136630862	0.217271508	-0.231151188	-0.270300137
0.069634148	-0.070061442	-0.198072758	-0.038043879	0.213131179	0.170888034
-0.090542074	-0.004331875	0.193431464	0.001222944	0.012126302	-0.16041197
0.045794103	0.116430182	-0.076515051	-0.073686072	-0.019802856	0.086991115
0.156551592	-0.125138134	-0.02444415	0.072743079	-0.013643932	-0.011492729

```
Ma trận chuyển vị của ma trận đã nhập là:
```

1.0	3.0	7.0	7.0	2.0	8.0
1.0	5.0	2.0	8.0	3.0	2.0
2.0	6.0	3.0	9.0	8.0	3.0
5.0	5.0	8.0	2.0	7.0	1.0
8.0	9.0	7.0	2.0	4.0	4.0
7.0	0.0	9.0	3.0	1.0	7.0

Ma trận cỡ 10x10:

- Nhập ma trận A từ tệp tin (*input_c.txt*):

```

Nhập tên tệp tin (filename.txt) cho ma trận A hoặc nhấn Enter để nhập từ bàn phím: input_c.txt
Ma trận A là:
1.0      1.0      2.0      5.0      8.0      7.0      9.0      1.0      4.0      7.0
3.0      5.0      6.0      5.0      9.0      0.0      7.0      6.0      3.0      2.0
7.0      2.0      3.0      8.0      7.0      9.0      3.0      0.0      1.0      1.0
7.0      8.0      9.0      2.0      2.0      3.0      4.0      1.0      7.0      9.0
2.0      3.0      8.0      7.0      4.0      1.0      6.0      6.0      2.0      0.0
8.0      2.0      3.0      1.0      4.0      7.0      9.0      6.0      3.0      1.0
4.0      8.0      6.0      2.0      5.0      1.0      0.0      2.0      1.0      5.0
5.0      5.0      6.0      3.0      4.0      1.0      8.0      7.0      9.0      2.0
5.0      6.0      4.0      2.0      8.0      3.0      7.0      1.0      2.0      5.0
7.0      9.0      6.0      3.0      2.0      1.0      5.0      8.0      3.0      1.0

```

- Nhập ma trận B từ tệp tin (*input_d.txt*):

```

Nhập tên tệp tin (filename.txt) cho ma trận B hoặc nhấn Enter để nhập từ bàn phím: input_d.txt
Ma trận B là:
1.0      4.0      7.0      5.0      3.0      2.0      8.0      9.0      1.0      6.0
2.0      8.0      6.0      7.0      9.0      3.0      2.0      4.0      7.0      8.0
5.0      3.0      1.0      7.0      8.0      9.0      6.0      1.0      0.0      7.0
1.0      4.0      2.0      3.0      8.0      6.0      5.0      1.0      7.0      3.0
1.0      8.0      6.0      4.0      7.0      3.0      5.0      8.0      1.0      0.0
9.0      6.0      4.0      3.0      7.0      8.0      2.0      1.0      4.0      7.0
2.0      2.0      3.0      4.0      7.0      8.0      5.0      6.0      1.0      4.0
8.0      6.0      3.0      1.0      7.0      5.0      2.0      4.0      1.0      0.0
1.0      8.0      9.0      6.0      3.0      7.0      2.0      4.0      0.0      5.0
8.0      2.0      4.0      7.0      5.0      6.0      3.0      1.0      1.0      2.0

```

- Tổng hai ma trận A và B:

```

Tổng hai ma trận là:
2.0      5.0      9.0      10.0      11.0      9.0      17.0      10.0      5.0      13.0
5.0      13.0      12.0      12.0      18.0      3.0      9.0      10.0      10.0      10.0
12.0      5.0      4.0      15.0      15.0      18.0      9.0      1.0      1.0      8.0
8.0      12.0      11.0      5.0      10.0      9.0      9.0      2.0      14.0      12.0
3.0      11.0      14.0      11.0      11.0      4.0      11.0      14.0      3.0      0.0
17.0      8.0      7.0      4.0      11.0      15.0      11.0      7.0      7.0      8.0
6.0      10.0      9.0      6.0      12.0      9.0      5.0      8.0      2.0      9.0
13.0      11.0      9.0      4.0      11.0      6.0      10.0      11.0      10.0      2.0
6.0      14.0      13.0      8.0      11.0      10.0      9.0      5.0      2.0      10.0
15.0      11.0      10.0      10.0      7.0      7.0      8.0      9.0      4.0      3.0

```

- Hiệu hai ma trận A và B:

```

Hiệu hai ma trận là:
0.0      -3.0      -5.0      0.0      5.0      5.0      1.0      -8.0      3.0      1.0
1.0      -3.0      0.0      -2.0      0.0      -3.0      5.0      2.0      -4.0      -6.0
2.0      -1.0      2.0      1.0      -1.0      0.0      -3.0      -1.0      1.0      -6.0
6.0      4.0      7.0      -1.0      -6.0      -3.0      0.0      0.0      0.0      6.0
1.0      -5.0      2.0      3.0      -3.0      -2.0      1.0      -2.0      1.0      0.0
-1.0      -4.0      -1.0      -2.0      -3.0      -1.0      7.0      5.0      -1.0      -6.0
2.0      6.0      3.0      -2.0      -2.0      -7.0      -5.0      -4.0      0.0      1.0
-3.0      -1.0      3.0      2.0      -3.0      -4.0      6.0      3.0      8.0      2.0
4.0      -2.0      -5.0      -4.0      5.0      -4.0      5.0      -3.0      2.0      0.0
-1.0      7.0      2.0      -4.0      -3.0      -5.0      2.0      7.0      2.0      -1.0

```

- Tích hai ma trận A và B:

Tích hai ma trận là:

175.0	214.0	195.0	204.0	290.0	280.0	177.0	172.0	96.0	162.0
138.0	240.0	195.0	209.0	315.0	251.0	199.0	210.0	97.0	162.0
137.0	211.0	180.0	174.0	268.0	225.0	191.0	170.0	124.0	185.0
194.0	249.0	248.0	299.0	317.0	301.0	215.0	190.0	105.0	265.0
130.0	186.0	136.0	169.0	278.0	239.0	173.0	146.0	92.0	154.0
172.0	215.0	201.0	182.0	270.0	252.0	191.0	214.0	77.0	190.0
123.0	182.0	155.0	190.0	232.0	168.0	142.0	134.0	90.0	158.0
158.0	262.0	239.0	237.0	309.0	291.0	201.0	221.0	86.0	209.0
138.0	216.0	201.0	218.0	281.0	229.0	188.0	201.0	94.0	181.0
154.0	236.0	201.0	213.0	300.0	234.0	187.0	200.0	111.0	209.0

- Tích ma trận A với một số:

Vui lòng chọn: 4
Nhập số muốn nhân với ma trận: 9
Tích của ma trận với một số là:

9.0	9.0	18.0	45.0	72.0	63.0	81.0	9.0	36.0	63.0
27.0	45.0	54.0	45.0	81.0	0.0	63.0	54.0	27.0	18.0
63.0	18.0	27.0	72.0	63.0	81.0	27.0	0.0	9.0	9.0
63.0	72.0	81.0	18.0	18.0	27.0	36.0	9.0	63.0	81.0
18.0	27.0	72.0	63.0	36.0	9.0	54.0	54.0	18.0	0.0
72.0	18.0	27.0	9.0	36.0	63.0	81.0	54.0	27.0	9.0
36.0	72.0	54.0	18.0	45.0	9.0	0.0	18.0	9.0	45.0
45.0	45.0	54.0	27.0	36.0	9.0	72.0	63.0	81.0	18.0
45.0	54.0	36.0	18.0	72.0	27.0	63.0	9.0	18.0	45.0
63.0	81.0	54.0	27.0	18.0	9.0	45.0	72.0	27.0	9.0

- Kết quả định thức, hạng ma trận A, ma trận nghịch đảo, ma trận chuyển vị của ma trận A:

Định thức của ma trận đã nhập là: 75371472.00000006

Hạng của ma trận đã nhập là: 10

Ma trận nghịch đảo của ma trận đã nhập là:

-0.096480934	0.48090442	0.08575279	0.284046489	-0.245636094	0.044656419	-0.336074477	-0.268033799	-0.172715666	0.106748399
0.035370345	-0.427510146	-0.017377092	-0.227832687	0.130996526	-0.115558391	0.193860483	0.21297229	0.266115819	0.065031449
-0.087517582	-0.236633736	-0.051158414	-0.065251757	0.24794332	0.080960632	0.210611543	0.115961952	0.115404844	-0.218651694
0.053378644	0.269599232	0.094854868	0.156858858	-0.106887219	-0.127056428	-0.280010904	-0.181344541	-0.14467515	0.193742216
-0.035379633	0.054741043	0.014086351	-0.074330179	-0.032211193	0.03570059	0.130866411	0.073794512	0.016960993	-0.129366029
0.063622321	-0.487626884	-0.016975601	-0.251415708	0.196999788	0.061073187	0.357896155	0.241227769	0.119284867	-0.11981934
0.007286086	-0.090001652	-0.042638215	-0.004762823	0.085356977	-0.028643596	-0.14570757	-0.018167683	0.190983944	0.053101417
0.075920555	0.233946181	-0.036537909	0.042353478	-0.116831525	0.094031015	0.030684395	-0.11087585	-0.289411835	0.077378242
-0.004001328	-0.134203801	0.030797939	-0.068700184	-0.008180575	-0.048563892	0.08726437	0.23836114	0.020004837	-0.080583805
0.097349047	0.451460109	-0.019586389	0.27426564	-0.226682703	0.027023248	-0.214218955	-0.318307104	-0.297282731	0.133931469

Ma trận chuyển vị của ma trận đã nhập là:

1.0	3.0	7.0	7.0	2.0	8.0	4.0	5.0	5.0	7.0
1.0	5.0	2.0	8.0	3.0	2.0	8.0	5.0	6.0	9.0
2.0	6.0	3.0	9.0	8.0	3.0	6.0	6.0	4.0	6.0
5.0	5.0	8.0	2.0	7.0	1.0	2.0	3.0	2.0	3.0
8.0	9.0	7.0	2.0	4.0	4.0	5.0	4.0	8.0	2.0
7.0	0.0	9.0	3.0	1.0	7.0	1.0	1.0	3.0	1.0
9.0	7.0	3.0	4.0	6.0	9.0	0.0	8.0	7.0	5.0
1.0	6.0	0.0	1.0	6.0	6.0	2.0	7.0	1.0	8.0
4.0	3.0	1.0	7.0	2.0	3.0	1.0	9.0	2.0	3.0
7.0	2.0	1.0	9.0	0.0	1.0	5.0	2.0	5.0	1.0

Ma trận với số thập phân:

- Nhập ma trận A từ tệp tin (*input_h.txt*):

```

Nhập tên tệp tin (filename.txt) cho ma trận A hoặc nhấn Enter để nhập từ bàn phím: input_h.txt
Ma trận A là:
21.212      4.923      17.856      5.697      -3.874      2.247      8.341      9.976      10.012      6.039
2.963       8.002      16.763      7.147      9.214      3.367      2.148      4.412      7.697      8.102
5.239      -3.107       1.297      7.1478     8.694      9.145     -6.123     -1.126       0.0       7.395
1.239       4.248       2.223     -3.125     8.295      6.301      5.221      1.015      7.377      3.399
1.239       8.147       6.952      4.452     -7.356     23.014     5.556      8.953      1.237      0.119
-9.391     -6.611      4.315     -3.678      7.111      8.394      2.146      1.234      4.142      7.822
2.321       2.361      3.117      4.264      7.328      8.642      5.742      6.915      1.837     -4.365
8.333       6.141      3.227      1.366     17.541      5.259      2.341      4.005      1.697      0.316
1.005       8.384      9.921      6.156     23.475      7.347      2.371      4.145      0.023     -5.671
-8.286       2.222      4.143      7.446      5.249      6.347      3.641      1.111      1.878      2.463

```

- Nhập ma trận B từ tệp tin (*input_k.txt*):

```

Nhập tên tệp tin (filename.txt) cho ma trận B hoặc nhấn Enter để nhập từ bàn phím: input_k.txt
Ma trận B là:
4.256      -3.192       7.891     -22.405      6.147      8.972      1.321      0.556      3.712     49.039
0.963      19.786     36.237      8.841      3.541      5.267      7.148      2.912     6.597     44.102
2.239      -1.107       5.297      3.978     26.694      4.145     -9.623     4.126     2.832      1.395
27.239      2.248     41.223     -4.125      0.295      9.301      5.721      2.215     36.377     13.399
3.239       6.147     31.952      9.452     -2.644      3.014      5.556      8.953      1.237     -5.119
-5.391     -6.611      4.315     -3.678      7.211      0.394      6.146      1.234      4.342      9.822
1.321       6.361      3.117      5.264      1.328      8.642      3.742      4.415      0.837     -4.365
-8.333       3.141      2.227     -1.866      7.541     25.259      2.341     -4.005      1.697      0.316
2.805       7.384      1.921      6.156      4.475      1.347      2.371      4.145      1.023     -5.671
-3.286       2.722     11.143     -2.446      7.249      5.347     33.641      1.111     -1.878      2.463

```

- Tổng hai ma trận A và B:

```

Tổng hai ma trận là:
25.468      1.731      25.747     -16.708      2.273     11.219      9.662     10.532     13.724     55.078
3.926      27.788       53.0     15.988     12.755      8.634      9.296      7.324     14.294     52.204
7.478     -4.214      6.594     11.1258     35.388     13.29     -15.746      3.0       2.832      8.79
28.478      6.496     43.446      -7.25      8.59     15.602     10.942      3.23     43.754     16.798
4.478      14.294     38.904     13.904     -10.0     26.028     11.112     17.906      2.474      -5.0
-14.782     -13.222      8.63     -7.356     14.322      8.788      8.292      2.468      8.484     17.644
3.642      8.722      6.234      8.656     17.284      9.484     11.33      2.674     -8.73
0.0         9.282      5.454     -0.5      25.082     30.518      4.682      0.0       3.394      0.632
3.81       15.768     11.842     12.312     27.95      8.694      4.742      8.29     1.046     -11.342
-11.572      4.944     15.286       5.0     12.498     11.694     37.282      2.222      0.0       4.926

```

- Hiệu hai ma trận A và B:

```

Hiệu hai ma trận là:
16.956      8.115      9.965      28.102     -10.021     -6.725      7.02      9.42      6.3     -43.0
2.0       -11.784     -19.474     -1.694      5.673     -1.9      -5.0       1.5       1.1     -36.0
3.0        -2.0       -4.0       3.1698     -18.0      5.0       3.5     -5.252     -2.832      6.0
-26.0       2.0      -39.0       1.0       8.0      -3.0     -0.5     -1.2     -29.0     -10.0
-2.0        2.0     -25.0      -5.0     -4.712     20.0      0.0      0.0      0.0      5.238
-4.0        0.0      0.0      0.0      0.0     -0.1      8.0     -4.0      0.0     -2.0
1.0         -4.0      0.0     -1.0      6.0      0.0      2.0      2.5      1.0      0.0
16.666       3.0      1.0     3.232     10.0     -20.0      0.0      8.01      0.0      0.0
-1.8        1.0      8.0      0.0     19.0      6.0      0.0      0.0     -1.0      0.0
-5.0       -0.5      -7.0     9.892     -2.0      1.0     -30.0      0.0      3.756      0.0

```

- Tích hai ma trận A và B:

Tích hai ma trận là:

201.645784	158.828126	695.864141	-356.926802	827.483434	702.296907	197.723312	125.591939	396.799995	1325.313633
225.257818	287.168524	1127.285595	146.907928	625.34506	417.480023	321.7519	229.310562	399.207832	571.581718
172.7617262	-92.9370706	608.3374134	-168.852139	137.874469	90.4888028	340.8594128	89.863841	292.1773986	307.375622
-69.919439	180.066375	409.476272	149.447966	176.820748	139.991997	231.294058	150.689446	-29.883727	172.827989
-62.129379	35.112015	430.574503	-80.751632	490.049053	387.608492	167.70211	19.431637	351.696393	724.962853
-180.610609	-56.252728	-74.443517	235.993804	162.083832	-13.540956	252.751928	89.598523	-161.36578	-762.502051
81.88291	93.281776	555.777876	40.468862	186.382877	314.753501	3.357163	107.164227	261.335771	282.748272
87.717293	208.324253	967.850931	35.772392	194.252297	313.498752	194.234706	204.070842	184.705399	646.340323
225.96655	274.098559	1353.310845	279.300819	286.823969	320.100138	4.439041	286.442541	391.7044	444.21541
154.483345	120.092787	584.005111	239.957486	141.085369	117.338065	179.375627	112.423014	302.777271	-187.449126

- Tích ma trận A với một số:

Vui lòng chọn: 4
Nhập số muốn nhân với ma trận: 10
Tích của ma trận với một số là:

212.12	49.23	178.56	56.97	-38.74	22.47	83.41	99.76	100.12	60.39
29.63	80.02	167.63	71.47	92.14	33.67	21.48	44.12	76.97	81.02
52.39	-31.07	12.97	71.478	86.94	91.45	-61.23	-11.26	0.0	73.95
12.39	42.48	22.23	-31.25	82.95	63.01	52.21	10.15	73.77	33.99
12.39	81.47	69.52	44.52	-73.56	230.14	55.56	89.53	12.37	1.19
-93.91	-66.11	43.15	-36.78	71.11	83.94	21.46	12.34	41.42	78.22
23.21	23.61	31.17	42.64	73.28	86.42	57.42	69.15	18.37	-43.65
83.33	61.41	32.27	13.66	175.41	52.59	23.41	40.05	16.97	3.16
10.05	83.84	99.21	61.56	234.75	73.47	23.71	41.45	0.23	-56.71
-82.86	22.22	41.43	74.46	52.49	63.47	36.41	11.11	18.78	24.63

Kết quả đã được lưu vào file: output.txt

- Kết quả định thức, hạng ma trận A, ma trận nghịch đảo, ma trận chuyển vị của ma trận A:

Định thức của ma trận đã nhập là: 35928445115.12984

Hạng của ma trận đã nhập là: 10

Ma trận nghịch đảo của ma trận đã nhập là:

0.045898475	-0.060437917	0.022984721	0.014814467	0.00883751	-0.010625107	-0.048543199	0.011040979	0.026842751	0.004878554
-0.041122746	0.057660504	-0.032624966	0.010112578	0.030649161	-0.048068453	-0.031409369	0.061693413	-0.034023853	0.004409959
0.053844986	-0.041185928	-0.008136931	-0.013325351	0.013812346	0.039278123	-0.101137227	-0.063927068	0.105938571	-0.006243833
0.002728606	0.007173064	0.036971682	-0.005792572	-0.024168883	-0.047334664	0.05400976	-0.025220678	-0.020764481	0.069339808
-0.004940172	-0.003828048	0.001317737	-0.004171554	-0.016247542	0.012690684	0.004073698	0.034399967	0.00702551	0.005969506
0.011398168	-0.044670535	0.03373051	0.041756038	0.031199508	0.005698103	-0.029879933	-0.042941342	0.043269451	-0.007323366
0.114703372	-0.189590966	-0.0549604	-0.005176224	0.001818718	0.031534792	-0.174554579	0.090890172	0.047696083	0.203146315
-0.106562942	0.193958821	-0.044926375	-0.12542967	-0.003230895	0.022655059	0.244694244	0.109965291	-0.172669861	-0.118573934
-0.0737527	0.142521693	0.059199586	0.150281748	-0.048210824	-0.062564066	0.231131547	-0.183219843	-0.043452332	-0.139027646
0.008584386	-0.000370901	-0.02468247	-0.068428044	0.014755053	0.03378856	-0.08838185	0.158348913	-0.080390171	0.084656945

Ma trận chuyển vị của ma trận đã nhập là:

21.212	2.963	5.239	1.239	1.239	-9.391	2.321	8.333	1.005	-8.286
4.923	8.002	-3.107	4.248	8.147	-6.611	2.361	6.141	8.384	2.222
17.856	16.763	1.297	2.223	6.952	4.315	3.117	3.227	9.921	4.143
5.697	7.147	7.1478	-3.125	4.452	-3.678	4.264	1.366	6.156	7.446
-3.874	9.214	8.694	8.295	-7.356	7.111	7.328	17.541	23.475	5.249
2.247	3.367	9.145	6.301	23.014	8.394	8.642	5.259	7.347	6.347
8.341	2.148	-6.123	5.221	5.556	2.146	5.742	2.341	2.371	3.641
9.976	4.412	-1.126	1.015	8.953	1.234	6.915	4.005	4.145	1.111
10.012	7.697	0.0	7.377	1.237	4.142	1.837	1.697	0.023	1.878
6.039	8.102	7.395	3.399	0.119	7.822	-4.365	0.316	-5.671	2.463

4.2 Kiểm thử các lỗi chương trình.

1. Lỗi nhập dữ liệu từ bàn phím.

- Nhập sai số hàng, số cột:

```
Nhập tên tệp tin (filename.txt) cho ma trận A hoặc nhấn Enter để nhập từ bàn phím:
Nhập số hàng của ma trận A: 2
Nhập số cột của ma trận A: -2
Số hàng và số cột phải là số nguyên dương. Vui lòng nhập lại.
```

- Nhập sai giá trị:

```
Nhập tên tệp tin (filename.txt) cho ma trận A hoặc nhấn Enter để nhập từ bàn phím:
Nhập số hàng của ma trận A: 2
Nhập số cột của ma trận A: 2
Nhập giá trị [1][1]: 1
Nhập giá trị [1][2]: -2
Nhập giá trị [2][1]: h
Giá trị không hợp lệ: các giá trị phải là số nguyên hoặc số thực. Vui lòng nhập lại.
Nhập giá trị [2][1]: 3
Nhập giá trị [2][2]: -5
```

2. Lỗi nhập dữ liệu từ tệp tin.

- Lỗi khai báo dòng đầu tiên trong file sai (*input1.txt*):

```
Nhập tên tệp tin (filename.txt) cho ma trận A hoặc nhấn Enter để nhập từ bàn phím: input1.txt
Traceback (most recent call last):
  File "c:\Users\admin\Downloads\BÁO CÁO CUỐI KỲ KTLT\matrix.py", line 60, in input_from_file
    raise ValueError("Dòng đầu tiên của file chỉ chứa hai giá trị nguyên dương (số hàng, số cột) và cách nhau bởi khoảng trắng.")
ValueError: Dòng đầu tiên của file chỉ chứa hai giá trị nguyên dương (số hàng, số cột) và cách nhau bởi khoảng trắng.
```

- Số lượng giá trị không tương ứng với số hàng, số cột (*input2.txt*):

```
Nhập tên tệp tin (filename.txt) cho ma trận A hoặc nhấn Enter để nhập từ bàn phím: input2.txt
Traceback (most recent call last):
  File "c:\Users\admin\Downloads\BÁO CÁO CUỐI KỲ KTLT\matrix.py", line 261, in <module>
    matrix_a.input_from_file(filename_a)
  File "c:\Users\admin\Downloads\BÁO CÁO CUỐI KỲ KTLT\matrix.py", line 68, in input_from_file
    raise ValueError("Số lượng giá trị không tương ứng với số cột của ma trận.")
ValueError: Số lượng giá trị không tương ứng với số cột của ma trận.
```


4. Lỗi sai khi tính tổng/hiệu/tích hai ma trận.

- Nhập hai ma trận không cùng kích thước từ bàn phím:

```

Nhập tên tệp tin (filename.txt) cho ma trận A hoặc nhấn Enter để nhập từ bàn phím:
Nhập số hàng của ma trận A: 2
Nhập số cột của ma trận A: 2
Nhập giá trị [1][1]: 1
Nhập giá trị [1][2]: 4
Nhập giá trị [2][1]: 7
Nhập giá trị [2][2]: 8
Ma trận A là:

      1.0      4.0
      7.0      8.0

Vui lòng chọn: 1
Nhập tên tệp tin (filename.txt) cho ma trận B hoặc nhấn Enter để nhập từ bàn phím:
Nhập số hàng của ma trận B: 3
Nhập số cột của ma trận B: 3
Nhập giá trị [1][1]: 2
Nhập giá trị [1][2]: 7
Nhập giá trị [1][3]: 6
Nhập giá trị [2][1]: 8
Nhập giá trị [2][2]: 2
Nhập giá trị [2][3]: 7
Nhập giá trị [3][1]: 8
Nhập giá trị [3][2]: 2
Nhập giá trị [3][3]: 6
Ma trận B là:

      2.0      7.0      6.0
      8.0      2.0      7.0
      8.0      2.0      6.0

Kết quả đã được lưu vào file: output.txt
Traceback (most recent call last):
  File "c:\Users\admin\Downloads\BÁO CÁO CUỐI KỲ KTLT\matrix.py", line 306, in <module>
    matrix_sum = matrix_a_copy.addition(matrix_b)
                  ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\Users\admin\Downloads\BÁO CÁO CUỐI KỲ KTLT\matrix.py", line 95, in addition
    raise ValueError("Hai ma trận không có kích thước giống nhau.")
ValueError: Hai ma trận không có kích thước giống nhau.

```

- Nhập hai ma trận không cùng kích thước từ tệp tin (*input4.txt*, *input5.txt*):

```

ms-python\python 20211218\python 110 (110\python\debugpy\adapter\7.7.1\python\python.exe)
Nhập tên tệp tin (filename.txt) cho ma trận A hoặc nhấn Enter để nhập từ bàn phím: input4.txt
Ma trận A là:

    1.0      1.0      2.0      5.0      8.0
    3.0      5.0      6.0      5.0      9.0
    7.0      2.0      3.0      6.0      7.0
    7.0      8.0      9.0      2.0      2.0
    2.0      3.0      8.0      7.0      4.0

Kết quả đã được lưu vào file: output.txt
Menu:
1. Cộng hai ma trận
2. Hiệu hai ma trận
3. Nhân hai ma trận
4. Nhân một số với ma trận
5. Tính định thức của ma trận
6. Tính hạng của ma trận
7. Tính ma trận nghịch đảo
8. Tính ma trận chuyển vị
9. Thoát khỏi chương trình

Vui lòng chọn: 1
Nhập tên tệp tin (filename.txt) cho ma trận B hoặc nhấn Enter để nhập từ bàn phím: input5.txt
Ma trận B là:

    1.0      1.0      2.0      5.0      4.0      7.0
    3.0      5.0      6.0      5.0      0.0      2.0
    7.0      2.0      3.0      1.0      1.0      8.0
    7.0      8.0      9.0      2.0      9.0      2.0
    2.0      3.0      8.0      7.0      7.0      3.0
    0.0      2.0      1.0      4.0      9.0      3.0

Kết quả đã được lưu vào file: output.txt
Traceback (most recent call last):
  File "c:\Users\admin\Downloads\BÁO CÁO CUỐI KỲ KTLT\matrix.py", line 306, in <module>
    matrix_sum = matrix_a_copy.addition(matrix_b)
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\Users\admin\Downloads\BÁO CÁO CUỐI KỲ KTLT\matrix.py", line 95, in addition
    raise ValueError("Hai ma trận không có kích thước giống nhau.")
ValueError: Hai ma trận không có kích thước giống nhau.

```

5. Lỗi sai khi tính định thức ma trận.

- Nhập ma trận không phải ma trận vuông từ tệp tin (*input6.txt*):

```

Nhập tên tệp tin (filename.txt) cho ma trận A hoặc nhấn Enter để nhập từ bàn phím: input6.txt
Ma trận A là:

      1.0      1.0      2.0      5.0      4.0      7.0
      3.0      5.0      6.0      5.0      0.0      2.0
      7.0      2.0      3.0      1.0      1.0      8.0
      7.0      8.0      9.0      2.0      9.0      2.0
      2.0      3.0      8.0      7.0      7.0      3.0

Kết quả đã được lưu vào file: output.txt
Menu:
1. Cộng hai ma trận
2. Hiệu hai ma trận
3. Nhân hai ma trận
4. Nhân một số với ma trận
5. Tính định thức của ma trận
6. Tính hạng của ma trận
7. Tính ma trận nghịch đảo
8. Tính ma trận chuyển vị
9. Thoát khỏi chương trình

Vui lòng chọn: 5
Traceback (most recent call last):
  File "c:\Users\admin\Downloads\BÁO CÁO CUỐI KỲ KTLT\matrix.py", line 333, in <module>
    det = matrix_a_copy.calculate_determinant()
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\Users\admin\Downloads\BÁO CÁO CUỐI KỲ KTLT\matrix.py", line 178, in calculate_determinant
    raise ValueError("Ma trận không phải ma trận vuông")
ValueError: Ma trận không phải ma trận vuông

```

- Nhập ma trận rỗng từ tệp tin (*input7.txt*):

```

Nhập tên tệp tin (filename.txt) cho ma trận A hoặc nhấn Enter để nhập từ bàn phím: input7.txt
Ma trận A là:

Kết quả đã được lưu vào file: output.txt
Menu:
1. Cộng hai ma trận
2. Hiệu hai ma trận
3. Nhân hai ma trận
4. Nhân một số với ma trận
5. Tính định thức của ma trận
6. Tính hạng của ma trận
7. Tính ma trận nghịch đảo
8. Tính ma trận chuyển vị
9. Thoát khỏi chương trình

Vui lòng chọn: 5
Định thức của ma trận đã nhập là:
Ma trận rỗng
Kết quả đã được lưu vào file: output.txt

```

- Nhập ma trận có hạng nhỏ hơn số hàng từ tệp tin (*input8.txt*):

```

Nhập tên tệp tin (filename.txt) cho ma trận A hoặc nhấn Enter để nhập từ bàn phím: input8.txt
Ma trận A là:

      1.0      0.0      0.0      0.0      0.0
      0.0      0.0      0.0      0.0      0.0
      0.0      0.0      1.0      0.0      0.0
      1.0      1.0      0.0      1.0      0.0
      1.0      1.0      0.0      0.0      1.0

Kết quả đã được lưu vào file: output.txt
Menu:
1. Cộng hai ma trận
2. Hiệu hai ma trận
3. Nhân hai ma trận
4. Nhân một số với ma trận
5. Tính định thức của ma trận
6. Tính hạng của ma trận
7. Tính ma trận nghịch đảo
8. Tính ma trận chuyển vị
9. Thoát khỏi chương trình

Vui lòng chọn: 5
Định thức của ma trận đã nhập là:
Định thức bằng 0

```

- Nhập ma trận không phải ma trận vuông từ bàn phím:

```

Nhập tên tệp tin (filename.txt) cho ma trận A hoặc nhấn Enter để nhập từ bàn phím:
Nhập số hàng của ma trận A: 2
Nhập số cột của ma trận A: 3
Nhập giá trị [1][1]: 1
Nhập giá trị [1][2]: 5
Nhập giá trị [1][3]: -9
Nhập giá trị [2][1]: 7
Nhập giá trị [2][2]: -2
Nhập giá trị [2][3]: 4
Ma trận A là:

      1.0      5.0     -9.0
      7.0     -2.0      4.0

Kết quả đã được lưu vào file: output.txt
Menu:
1. Cộng hai ma trận
2. Hiệu hai ma trận
3. Nhân hai ma trận
4. Nhân một số với ma trận
5. Tính định thức của ma trận
6. Tính hạng của ma trận
7. Tính ma trận nghịch đảo
8. Tính ma trận chuyển vị
9. Thoát khỏi chương trình

Vui lòng chọn: 5
Traceback (most recent call last):
  File "c:\Users\admin\Downloads\BÁO CÁO CUỐI KỲ KTLT\matrix.py", line 333, in <module>
    det = matrix_a_copy.calculate_determinant()
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\Users\admin\Downloads\BÁO CÁO CUỐI KỲ KTLT\matrix.py", line 178, in calculate_determinant
    raise ValueError("Ma trận không phải ma trận vuông")
ValueError: Ma trận không phải ma trận vuông

```


- Nhập ma trận có hạng nhỏ hơn số hàng từ bàn phím:

```

Nhập tên tệp tin (filename.txt) cho ma trận A hoặc nhấn Enter để nhập từ bàn phím:
Nhập số hàng của ma trận A: 3
Nhập số cột của ma trận A: 3
Nhập giá trị [1][1]: 1
Nhập giá trị [1][2]: 0
Nhập giá trị [1][3]: 0
Nhập giá trị [2][1]: 0
Nhập giá trị [2][2]: 0
Nhập giá trị [2][3]: 0
Nhập giá trị [3][1]: 0
Nhập giá trị [3][2]: 1
Nhập giá trị [3][3]: 0
Ma trận A là:

      1.0      0.0      0.0
      0.0      0.0      0.0
      0.0      1.0      0.0

Kết quả đã được lưu vào file: output.txt
Menu:
1. Cộng hai ma trận
2. Hiệu hai ma trận
3. Nhân hai ma trận
4. Nhân một số với ma trận
5. Tính định thức của ma trận
6. Tính hạng của ma trận
7. Tính ma trận nghịch đảo
8. Tính ma trận chuyển vị
9. Thoát khỏi chương trình

Vui lòng chọn: 5
Định thức của ma trận đã nhập là:
0.0

```

6. Kiểm thử phép tính hạng ma trận.

- Nhập ma trận rỗng (*input7.txt*):

```

Nhập tên tệp tin (filename.txt) cho ma trận A hoặc nhấn Enter để nhập từ bàn phím: input7.txt
Ma trận A là:

Kết quả đã được lưu vào file: output.txt
Menu:
1. Cộng hai ma trận
2. Hiệu hai ma trận
3. Nhân hai ma trận
4. Nhân một số với ma trận
5. Tính định thức của ma trận
6. Tính hạng của ma trận
7. Tính ma trận nghịch đảo
8. Tính ma trận chuyển vị
9. Thoát khỏi chương trình

Vui lòng chọn: 6
Hạng của ma trận đã nhập là:
0

```

7. Kiểm thử ma trận nghịch đảo.

- Nhập ma trận không phải ma trận vuông từ tệp tin (*input6.txt*):

```

Nhập tên tệp tin (filename.txt) cho ma trận A hoặc nhấn Enter để nhập từ bàn phím: input6.txt
Ma trận A là:

      1.0      1.0      2.0      5.0      4.0      7.0
      3.0      5.0      6.0      5.0      0.0      2.0
      7.0      2.0      3.0      1.0      1.0      8.0
      7.0      8.0      9.0      2.0      9.0      2.0
      2.0      3.0      8.0      7.0      7.0      3.0

Kết quả đã được lưu vào file: output.txt
Menu:
1. Cộng hai ma trận
2. Hiệu hai ma trận
3. Nhân hai ma trận
4. Nhân một số với ma trận
5. Tính định thức của ma trận
6. Tính hạng của ma trận
7. Tính ma trận nghịch đảo
8. Tính ma trận chuyển vị
9. Thoát khỏi chương trình

Vui lòng chọn: 7
Traceback (most recent call last):
  File "c:\Users\admin\Downloads\BÁO CÁO CUỐI KỲ KTLT\matrix.py", line 347, in <module>
    inverse = matrix_a_copy.calculate_inverse()
    ~~~~~
  File "c:\Users\admin\Downloads\BÁO CÁO CUỐI KỲ KTLT\matrix.py", line 208, in calculate_inverse
    raise ValueError("Ma trận không phải ma trận vuông")
ValueError: Ma trận không phải ma trận vuông

```

- Nhập ma trận không thể nghịch đảo từ tệp tin (*input9.txt*):

```

Nhập tên tệp tin (filename.txt) cho ma trận A hoặc nhấn Enter để nhập từ bàn phím: input9.txt
Ma trận A là:

      1.0      3.0      5.0      7.0      -9.0
      2.0      6.0      10.0     14.0     -18.0
      3.0      9.0      15.0     21.0     -27.0
      4.0     12.0     20.0     28.0     -36.0
      5.0     15.0     25.0     35.0     -45.0

Kết quả đã được lưu vào file: output.txt
Menu:
1. Cộng hai ma trận
2. Hiệu hai ma trận
3. Nhân hai ma trận
4. Nhân một số với ma trận
5. Tính định thức của ma trận
6. Tính hạng của ma trận
7. Tính ma trận nghịch đảo
8. Tính ma trận chuyển vị
9. Thoát khỏi chương trình

Vui lòng chọn: 7
Traceback (most recent call last):
  File "c:\Users\admin\Downloads\BÁO CÁO CUỐI KỲ KTLT\matrix.py", line 347, in <module>
    inverse = matrix_a_copy.calculate_inverse()
    ~~~~~
  File "c:\Users\admin\Downloads\BÁO CÁO CUỐI KỲ KTLT\matrix.py", line 228, in calculate_inverse
    raise ValueError("Ma trận không thể nghịch đảo")
ValueError: Ma trận không thể nghịch đảo

```


- Nhập ma trận không phải ma trận vuông từ bàn phím:

```

Nhập tên tệp tin (filename.txt) cho ma trận A hoặc nhấn Enter để nhập từ bàn phím:
Nhập số hàng của ma trận A: 3
Nhập số cột của ma trận A: 2
Nhập giá trị [1][1]: 1
Nhập giá trị [1][2]: 2
Nhập giá trị [2][1]: -6
Nhập giá trị [2][2]: 2
Nhập giá trị [3][1]: -3
Nhập giá trị [3][2]: 9
Ma trận A là:

      1.0      2.0
     -6.0      2.0
     -3.0      9.0

Kết quả đã được lưu vào file: output.txt
Menu:
1. Cộng hai ma trận
2. Hiệu hai ma trận
3. Nhân hai ma trận
4. Nhân một số với ma trận
5. Tính định thức của ma trận
6. Tính hạng của ma trận
7. Tính ma trận nghịch đảo
8. Tính ma trận chuyển vị
9. Thoát khỏi chương trình

Vui lòng chọn: 7
Traceback (most recent call last):
  File "c:\Users\admin\Downloads\BÁO CÁO CUỐI KỲ KTLT\matrix.py", line 347, in <module>
    inverse = matrix_a_copy.calculate_inverse()
              ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\Users\admin\Downloads\BÁO CÁO CUỐI KỲ KTLT\matrix.py", line 208, in calculate_inverse
    raise ValueError("Ma trận không phải ma trận vuông")
ValueError: Ma trận không phải ma trận vuông

```

- Nhập ma trận không thể nghịch đảo từ bàn phím:

```

Nhập tên tệp tin (filename.txt) cho ma trận A hoặc nhấn Enter để nhập từ bàn phím:
Nhập số hàng của ma trận A: 3
Nhập số cột của ma trận A: 3
Nhập giá trị [1][1]: 1
Nhập giá trị [1][2]: 2
Nhập giá trị [1][3]: 3
Nhập giá trị [2][1]: 4
Nhập giá trị [2][2]: 5
Nhập giá trị [2][3]: 6
Nhập giá trị [3][1]: 7
Nhập giá trị [3][2]: 8
Nhập giá trị [3][3]: 9
Ma trận A là:

      1.0      2.0      3.0
      4.0      5.0      6.0
      7.0      8.0      9.0

Kết quả đã được lưu vào file: output.txt
Menu:
1. Cộng hai ma trận
2. Hiệu hai ma trận
3. Nhân hai ma trận
4. Nhân một số với ma trận
5. Tính định thức của ma trận
6. Tính hạng của ma trận
7. Tính ma trận nghịch đảo
8. Tính ma trận chuyển vị
9. Thoát khỏi chương trình

Vui lòng chọn: 7
Traceback (most recent call last):
  File "c:\Users\admin\Downloads\BÁO CÁO CUỐI KỲ KTLT\matrix.py", line 347, in <module>
    inverse = matrix_a_copy.calculate_inverse()
              ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\Users\admin\Downloads\BÁO CÁO CUỐI KỲ KTLT\matrix.py", line 228, in calculate_inverse
    raise ValueError("Ma trận không thể nghịch đảo")
ValueError: Ma trận không thể nghịch đảo

```

8. Kiểm thử Menu.

- Người dùng nhập sai lựa chọn:

```

Nhập tên tệp tin (filename.txt) cho ma trận A hoặc nhấn Enter để nhập từ bàn phím: input_a.txt
Ma trận A là:

    1.0      1.0      2.0      5.0      8.0      7.0
    3.0      5.0      6.0      5.0      9.0      0.0
    7.0      2.0      3.0      8.0      7.0      9.0
    7.0      8.0      9.0      2.0      2.0      3.0
    2.0      3.0      8.0      7.0      4.0      1.0
    8.0      2.0      3.0      1.0      4.0      7.0

Kết quả đã được lưu vào file: output.txt
Menu:
1. Cộng hai ma trận
2. Hiệu hai ma trận
3. Nhân hai ma trận
4. Nhân một số với ma trận
5. Tính định thức của ma trận
6. Tính hạng của ma trận
7. Tính ma trận nghịch đảo
8. Tính ma trận chuyển vị
9. Thoát khỏi chương trình

Vui lòng chọn: 10
Lựa chọn không hợp lệ, vui lòng chọn lại.

```

- Người dùng muốn thoát khỏi chương trình:

```

Nhập tên tệp tin (filename.txt) cho ma trận A hoặc nhấn Enter để nhập từ bàn phím: input_a.txt
Ma trận A là:

    1.0      1.0      2.0      5.0      8.0      7.0
    3.0      5.0      6.0      5.0      9.0      0.0
    7.0      2.0      3.0      8.0      7.0      9.0
    7.0      8.0      9.0      2.0      2.0      3.0
    2.0      3.0      8.0      7.0      4.0      1.0
    8.0      2.0      3.0      1.0      4.0      7.0

Kết quả đã được lưu vào file: output.txt
Menu:
1. Cộng hai ma trận
2. Hiệu hai ma trận
3. Nhân hai ma trận
4. Nhân một số với ma trận
5. Tính định thức của ma trận
6. Tính hạng của ma trận
7. Tính ma trận nghịch đảo
8. Tính ma trận chuyển vị
9. Thoát khỏi chương trình

Vui lòng chọn: 9
Bạn đã thoát khỏi chương trình.

```

(Các file input phục vụ cho quá trình kiểm thử đều đã được gửi trên Assignments Teams)

5 Tổng kết

Kết luận: Vậy để có thể trau dồi, tích lũy cho bản thân thêm các kỹ thuật lập trình để tạo ra một "chương trình tốt" chúng ta cần:

- Hiểu rõ nội dung chủ đề.
- Có tư duy và phương pháp lập trình.
- Hiểu sâu về máy tính.
- Nắm vững ngôn ngữ, nền tảng.
- Luôn rèn luyện kỹ năng.

Trong quá trình thực hiện, do kiến thức còn hạn hẹp và kinh nghiệm lập trình còn ít, em nhận thấy bài báo cáo vẫn còn nhiều lỗi sai, chương trình có một vài đoạn mã chưa thực sự tối ưu, giao diện chưa thân thiện với người dùng...

Tuy nhiên, việc thực hành bài tập lớn cũng là cơ hội tốt để em rèn luyện khả năng lập trình, học hỏi và tiếp thu kiến thức hơn từ chính những lỗi sai trong quá trình làm bài.

Cuối cùng, một lần nữa em xin gửi lời cảm ơn đến thầy **Vũ Thành Nam** vì đã dành thời gian xem đến những trang cuối cùng của bài báo cáo này. Chúc thầy thật nhiều sức khỏe để luôn tiếp tục cống hiến và truyền lửa nhiệt huyết tới các thế hệ sinh viên Bách Khoa!

Tài liệu tham khảo

- [1] Bài giảng Kỹ thuật lập trình của TS.Vũ Thành Nam, Viện Toán ứng dụng và Tin học, Đại Học Bách Khoa Hà Nội.
- [2] Cuốn sách "Python cơ bản" của tác giả Võ Tuấn Duy.
- [3] Cuốn sách "Think Python 2nd Edition" của Allen B. Downey.
- [4] <https://tailieuhust.com/tai-lieu-mon-ky-thuat-lap-trinh-hust>
- [5] <https://howkteam.vn/course/lap-trinh-python-co-ban>