

KIỂM TRA GIỮA KÌ

Học phần: Tin học cơ sở 4 Năm học: 2019 - 2020

Thời gian làm bài: 90 phút

Đề số 1

Câu 1. Xây dựng phương thức giải hệ phương trình bậc nhất 2 ẩn:

$$\begin{cases} ax + by = c \\ dx + ey = f \end{cases}$$

Chương trình đã được lên khung sẵn. Yêu cầu sinh viên hoàn thiện phương thức sau trong tệp **EqSystem.java**:

```
public static double[] solver(double a, double b, double c,
double d, double e, double f)
```

Phương thức này trả về nghiệm của hệ phương trình là một mảng **result** gồm 2 phần tử **double**. Xử lý một số trường hợp đặc biệt như sau:

- Nếu hệ phương trình vô nghiệm, thì hai phần tử **result[0]** và **result[1]** nhận giá trị **Double.NaN** (NaN là Not a Number, giá trị thuộc kiểu số thực được coi không phải là số)
- Nếu hệ phương trình có vô số nghiệm thì hai phần tử của mảng **result** nhận giá trị **Double.POSITIVE_INFINITY** (POSITIVE_INFINITY là giá trị kiểu số thực được coi là dương vô cùng)

Câu 2. Xây dựng chương trình tính giá trị **cosh(x)** theo khai triển Taylor.

$$\cosh(x) = \sum_{n=0}^{\infty} \frac{x^{2n}}{(2n)!} = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \dots$$

Chương trình đã được lên khung sẵn. Yêu cầu sinh viên hoàn thiện phương thức **public static double cosh(double x)** trong tệp **Trigonometry.java**. Phương thức này thực hiện tính và trả lại giá trị của **cosh(x)** theo công thức khai triển Taylor với **n** chạy tới 1000.

Câu 3. Sinh viên lớp THCS4 nhận được yêu cầu viết chương trình giải quyết bài toán sau. Cho bảng số liệu giá cổ phiếu niêm yết của **m** công ty (công ty thứ 1 đến công ty thứ **m**) vào giờ mở cửa hàng ngày của **n** ngày trong tháng (ngày thứ 1 đến ngày thứ **n**). Một người có nhu cầu khảo sát giá cổ phiếu của ngày thứ hai trong tháng, muốn chương trình làm được các việc sau:

- Giá cổ phiếu trung bình của các công ty trong ngày thứ hai;
- Giá cổ phiếu cao nhất trong ngày thứ hai;
- Giá cổ phiếu trong ngày thứ nhất của các công ty có cổ phiếu niêm yết cao nhất trong ngày thứ hai;
- Sắp xếp lại bảng số liệu theo thứ tự tăng dần của giá cổ phiếu trong ngày thứ hai.

Dữ liệu được biểu diễn dưới dạng ma trận, trong đó mỗi hàng là giá cổ phiếu niêm yết từ ngày thứ nhất đến ngày thứ n của mỗi công ty. Khi đó các yêu cầu bài toán trên được đưa về bài toán xử lý dữ liệu trên ma trận như sau:

- Tính giá trị trung bình ở cột thứ 2;
- Tìm giá trị lớn nhất ở cột thứ 2;
- In ra các giá trị ở cột 1 có giá trị tương ứng ở cột thứ 2 bằng giá trị lớn nhất;
- Sắp xếp các hàng của ma trận theo giá trị tăng dần của cột thứ 2.

Chương trình đã được lên khung sẵn. Yêu cầu sinh viên hoàn thiện các phương thức trong tệp **Matrix.java** theo các mô tả dưới đây:

- Phương thức **public static double calAverage(int[][] a)** thực hiện tính và trả lại giá trị trung bình của các phần tử ở cột thứ 2 của ma trận (chú ý cột thứ 2 tương ứng với chỉ số 1 trong mảng)
- Phương thức **public static int findMax(int[][] a)** thực hiện tìm và trả lại giá trị lớn nhất trong các phần tử ở cột thứ 2 của ma trận.
- Phương thức **public static void printMax(int[][] a)** thực hiện in ra các giá trị ở cột 1 có giá trị tương ứng ở cột thứ 2 bằng giá trị lớn nhất tìm được ở trên, mỗi giá trị in trên một dòng lần lượt theo thứ tự hàng.
- Phương thức **public static void sort(int[][] a)** Thực hiện việc sắp xếp lại các hàng của ma trận theo giá trị tăng dần của cột thứ 2.

Chú ý: Phương thức **public static void outputMatrix(int[][] a)** được viết sẵn, sinh viên không chỉnh sửa, thay đổi phương thức này.

Ví dụ ma trận nhập vào là:

1 2 3 1

1 5 2 4

6 1 0 9

Kết quả tìm được sẽ là:

- Trung bình cộng cột 2: 2.66
- Phần tử lớn nhất cột 2: 5
- Các phần tử ở cột 1 tương ứng với phần tử lớn nhất cột 2: 1
- Đổi chỗ các hàng theo thứ tự tăng dần cột 2 là:

6 1 0 9

1 2 3 1

1 5 2 4

Chú ý: Sinh viên đọc cẩn thận các lưu ý sau đây để tránh mắc các lỗi khi làm bài

1. Sinh viên chỉ cần hoàn thiện các phương thức được yêu cầu, tuyệt đối không chỉnh sửa nguyên mẫu (bao gồm danh sách đối và kiểu dữ liệu trả về) của các phương thức này, chỉ viết thêm vào phần nội dung phương thức.
2. Sinh viên có thể viết thêm các phương thức phụ trợ khác nếu cần, tuy nhiên tốt nhất tránh đặt tên trùng với tên các phương thức đã cho.
3. Mỗi phương thức được chấm điểm độc lập, đúng phương thức nào sẽ được điểm phương thức đó, tuy nhiên cần đảm bảo bài làm có thể biên dịch và chạy được. Các bài có lỗi biên dịch, hoặc không chạy được sẽ nhận điểm 0.
4. Các phương thức sinh viên chưa làm đến, hoặc không hoàn thiện được thì để lại như nguyên mẫu ban đầu, không chỉnh sửa, thêm bớt, tránh việc bài không biên dịch được.
5. Khi làm bài sinh viên có thể tự nộp bài và chấm điểm trên 1 bộ test mẫu, kết quả này chỉ là kết quả tham khảo, bài làm của sinh viên sẽ được chấm lại trên một bộ test khác và thông báo kết quả sau.

KIỂM TRA GIỮA KÌ

Học phần: Tin học cơ sở 4 Năm học: 2019 - 2020

Thời gian làm bài: 90 phút

Đề số 2

Câu 1. Xây dựng chương trình giải và biện luận phương trình bậc 2

$$ax^2 + bx + c = 0$$

Chương trình đã được lên khung sẵn. Yêu cầu sinh viên hoàn thiện phương thức sau trong tệp **QuadraticEq.java**:

public static double[] solver(double a, double b, double c)

Phương thức này trả về kết quả là một mảng **result** gồm 2 phần tử, tương ứng với 2 nghiệm của phương trình là **x1** và **x2** (**x1 ≤ x2**). Xử lý một số trường hợp đặc biệt như sau:

- Nếu phương trình có 1 nghiệm thì hai phần tử **result[0]** và **result[1]** bằng nhau và bằng nghiệm của phương trình
- Nếu phương trình vô nghiệm thì hai phần tử của mảng **result** đều nhận giá trị **Double.NaN** (NaN là Not a Number, giá trị thuộc kiểu số thực được coi không phải là số)
- Nếu phương trình có vô số nghiệm thì hai phần tử của mảng **result** đều nhận giá trị **Double.POSITIVE_INFINITY** (POSITIVE_INFINITY là giá trị kiểu số thực được coi là dương vô cùng)

Câu 2. Viết chương trình tính giá trị **sinh(x)** theo khai triển Taylor.

$$\sinh(x) = \sum_{n=0}^{\infty} \frac{x^{2n+1}}{(2n+1)!} = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \dots$$

Chương trình đã được lên khung sẵn. Yêu cầu sinh viên hoàn thiện phương thức

public static double sinh(double x) trong tệp **Trigonometry.java**

Phương thức này thực hiện tính và trả lại giá trị của tệp theo công thức khai triển Taylor với **n** chạy tới 1000.

Câu 3. Sinh viên lớp THCS4 nhận được yêu cầu viết chương trình giải quyết bài toán sau. Cho bảng số liệu giá cổ phiếu niêm yết của **m** công ty (công ty thứ 1 đến công ty thứ **m**) vào giờ mở cửa hàng ngày của **n** ngày trong một tháng nào đó (ngày thứ 1 đến ngày thứ **n**). Một người có nhu cầu khảo sát giá cổ phiếu của công ty thứ 3 trong danh sách công ty, muốn chương trình làm được các việc sau:

- Giá cổ phiếu trung bình của công ty thứ ba trong tháng;
- Giá cổ phiếu thấp nhất trong tháng của công ty đó;
- Giá cổ phiếu của công ty thứ nhất trong những ngày mà công ty thứ ba có giá cổ phiếu niêm yết thấp nhất;
- Sắp xếp lại bảng số liệu theo thứ tự giảm dần của giá cổ phiếu niêm yết trong tháng của công ty thứ ba.

Dữ liệu được biểu diễn dưới dạng ma trận, trong đó mỗi hàng là giá cổ phiếu niêm yết từ ngày thứ nhất đến ngày thứ n của mỗi công ty. Khi đó các yêu cầu bài toán trên được đưa về bài toán xử lý dữ liệu trên ma trận như sau:

- Tính giá trị trung bình ở hàng thứ 3;
- Tìm giá trị nhỏ nhất ở hàng thứ 3;
- In ra các giá trị ở hàng thứ nhất có giá trị tương ứng ở hàng thứ 3 bằng giá trị nhỏ nhất ;
- Sắp xếp các cột của ma trận theo giá trị giảm dần của hàng thứ 3.

Chương trình đã được lên khung sẵn. Yêu cầu sinh viên hoàn thiện các phương thức trong tệp **Matrix.java** theo các mô tả dưới đây:

- Phương thức **public static double calAverage(int[][] a)** thực hiện tính và trả lại giá trị trung bình của các phần tử ở hàng thứ 3 của ma trận (chú ý hàng thứ 3 tương ứng với chỉ số 2 trong mảng)
- Phương thức **public static int findMin(int[][] a)** thực hiện tìm và trả lại giá trị nhỏ nhất trong các phần tử ở hàng thứ 3 của ma trận.
- Phương thức **public static void printMin(int[][] a)** thực hiện in ra các giá trị ở hàng thứ nhất có giá trị tương ứng ở hàng thứ 3 bằng giá trị nhỏ nhất tìm được ở trên, mỗi giá trị in trên một dòng lần lượt theo thứ tự cột.
- Phương thức **public static void sort(int[][] a)** Thực hiện việc sắp xếp lại các cột của ma trận theo giá trị tăng dần của hàng thứ 3.

Chú ý: Phương thức **public static void outputMatrix(int[][] a)** được viết sẵn, sinh viên không chỉnh sửa, thay đổi phương thức này.

Ví dụ ma trận nhập vào là:

1 2 3 1

1 5 2 4

6 1 0 9

Kết quả tìm được sẽ là:

- Trung bình cộng hàng thứ 3: 4.00
- Phần tử nhỏ nhất hàng 3: 0
- Các phần tử ở hàng 1 tương ứng với phần tử nhỏ nhất hàng 3: 3
- Đổi chỗ các cột theo thứ tự giảm dần hàng 3 là:

1 1 2 3

4 1 5 2

9 6 1 0

Chú ý: Sinh viên đọc cẩn thận các lưu ý sau đây để tránh mắc các lỗi khi làm bài

6. Sinh viên chỉ cần hoàn thiện các phương thức được yêu cầu, tuyệt đối không chỉnh sửa nguyên mẫu (bao gồm danh sách đối và kiểu dữ liệu trả về) của các phương thức này, chỉ viết thêm vào phần nội dung phương thức.
7. Sinh viên có thể viết thêm các phương thức phụ trợ khác nếu cần, tuy nhiên tốt nhất tránh đặt tên trùng với tên các phương thức đã cho.
8. Mỗi phương thức được chấm điểm độc lập, đúng phương thức nào sẽ được điểm phương thức đó, tuy nhiên cần đảm bảo bài làm có thể biên dịch và chạy được. Các bài có lỗi biên dịch, hoặc không chạy được sẽ nhận điểm 0.
9. Các phương thức sinh viên chưa làm đến, hoặc không hoàn thiện được thì để lại như nguyên mẫu ban đầu, không chỉnh sửa, thêm bớt, tránh việc bài không biên dịch được.
10. Khi làm bài sinh viên có thể tự nộp bài và chấm điểm trên 1 bộ test mẫu, kết quả này chỉ là kết quả tham khảo, bài làm của sinh viên sẽ được chấm lại trên một bộ test khác và thông báo kết quả sau.