

KIỂM TRA GIỮA KỲ
Môn Tin học cơ sở 4 Thời
gian 80 phút
Đề số 1

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 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 nên 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 sau.**

Câu 1 (4 điểm)

Viết chương trình nhập vào số nguyên n ($n \geq 5$), sau đó nhập vào mảng a chứa n số nguyên.

- Kiểm tra xem dãy số vừa nhập có phải là dãy tăng không? Nếu có in ra YES, nếu không in ra NO.
- Tìm và in ra tổng: $a[0] - a[1] + a[2] - a[3] + \dots$

Dữ liệu được nhập từ bàn phím. Để thực hiện chương trình trên, sinh viên cần hoàn thiện các phương thức (hàm) trong file Mid1.java theo các yêu cầu sau:

1. Phương thức **public static void inputArray(int[] a, Scanner reader)** thực hiện việc nhập n số nguyên vào mảng a .
2. Phương thức **public static void printArray(int[] a)** thực hiện in ra n số nguyên có trong mảng a .
3. Phương thức **public static boolean isIncreasingSequence(int[] a)** thực hiện kiểm tra dãy a có phải là dãy tăng không.
4. Phương thức **public static int getSum(int[] a)** thực hiện việc tính tổng $a[0] - a[1] + a[2] - a[3] + \dots$

Câu 2 (6 điểm)

Viết chương trình nhập vào số nguyên n ($n \geq 3$), nhập vào ma trận số nguyên a kích thước $n \times n$.

- Tìm số nguyên tố nhỏ nhất có trong ma trận a

- Tìm và in ra số lớn nhất trong mỗi hàng của ma trận a
- Kiểm tra xem cột thứ 3 trong ma trận a gồm toàn các số âm không? Nếu có in ra YES, nếu không in ra NO

Dữ liệu được nhập từ bàn phím. Để thực hiện chương trình này, sinh viên cần hoàn thiện các phương thức (hàm) trong file Mid2.java theo các yêu cầu sau:

1. Phương thức **public static void** inputMatrix(**int**[][] a, Scanner reader) thực hiện nhập vào n*n số nguyên cho ma trận a.
2. Phương thức **public static int** getMinPrime(**int**[][] a) thực hiện tìm và trả lại số nguyên tố nhỏ nhất có trong ma trận a, nếu a không có số nguyên tố nào thì trả lại giá trị -1.
3. Phương thức **public static int** getMaxElement(**int**[][] a, **int** i) thực hiện tìm và trả lại giá trị lớn nhất nhất trên hàng i của ma trận a.
4. Phương thức **public static void** getMaxAllRow(**int**[][] a) thực hiện tìm và in ra giá trị lớn nhất trên mỗi hàng của ma trận (mỗi giá trị cách nhau một dấu cách “ ”).
5. Phương thức **public static boolean** negativeColumn(**int**[][] a, **int** k) thực hiện kiểm tra cột thứ 3 trong ma trận a có gồm toàn số âm không.