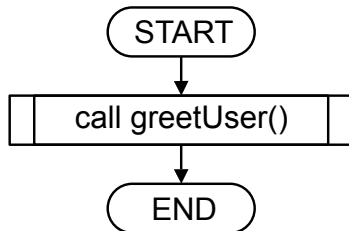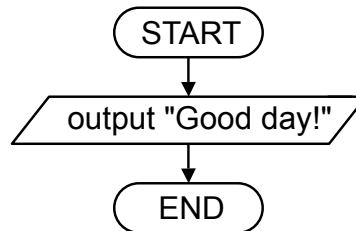1. The flowcharts below represent an algorithm to greet a user. The algorithm contains a subprocess. Convert the algorithm to Java directly – the main process will be the Java `main` method, and the subprocess must be a method named `greetUser`.

main process

START

call greetUser()

END

subprocess greetUser

START

output "Good day!"

END

2. Complete this question in the space on the reverse of this paper. Label the question.

   a) In the space on the back of the page, draw a flowchart that represents a process for printing the full name of a user. The `main` process should call a subprocess named `printFullName` that accepts two inputs, `firstName` and `lastName`, and prints them together as a full a name with the first name first and the last name after, with a space between the names.

   b) Translate the flowchart into Java code. Create a method `printFullName` that takes two input parameters of type `String` (one labeled `firstName` and the other labeled `lastName`), and prints the full name. (Note: the method should not return a value.)

3. Complete this question in the space on the reverse of this paper. Label the question.

   a) Draw a flowchart that represents a process to calculate the area of a rectangle. The main process should call a subprocess named `calculateArea` that accepts two numerical inputs, called `length` and `width`, and returns the area of the rectangle (length × width).

   b) Translate the flowchart into Java code. Create a method `calculateArea` that returns an integer and is called in the main method. The main method should print the area. Note: you can "*hard code*" the length and width in the main method – you do not need to get input from the user.