Lab Exercise #1: Task Master

In this laboratory exercise we define a **task** to be an activity that has a name, an execution time that specifies the amount of time needed to finish the task, and a deadline, which is the time the tasks is expected to finish. You are to create a simple task processing simulator that takes as input a list of tasks, executes them, and finally outputs a report.

You are also to implement a task as a class named simply as **Task**. You are to decide the needed properties of the class, but it must have the following methods:

Method	Description
void setName(String name)	Sets the name of the task
String getName()	Returns the name of the task
void setExecutionTime(int time)	Sets the amount of time needed by the task
int getExecutionTime()	Returns the amount of time needed by the task
void setDeadline(int time)	Sets deadline of the task
int getDeadline()	Returns the deadline of the task
void execute(int time)	Starts the execution of the task at the given time
int getStartTime()	Returns the time when the task started executing. If the task has not been executed yet, this method returns -1
int getEndTime()	Returns the time when the task finished executing. If the task has not been executed yet, this method returns -1
boolean delayed()	Returns true if the task finished its execution on or before the given deadline otherwise, returns false. This method also returns false if the task has not been executed yet.

When the simulator executes, it will first ask for the number of tasks needed then proceeds on asking for the details of each task. Once all the needed details are given, the simulator will process all the tasks then outputs the report.

For simplicity the time inside the simulator will be a non-negative integer that starts at 0. The execution time and deadline are positive integers. Time inside the simulator changes after every task execution.

The report contains the list of tasks in order of execution, the time when the tasks was started, the time the tasks ended, the deadline, and Yes or No indicating whether the tasks has missed its deadline.

Your submission must contain two files – the similar app and the Task class. You can safely assume that all inputs are correctly specified.

Sample Execution of the Simulator

```
Number of tasks: 3

Task 1: A
Execution: 5
Deadline: 7

Task 2: B
Execution: 7
Deadline: 10

Task 1: C
Execution: 3
Deadline: 15

Tasks      Start      End        Deadline   Missed
A          0          5          7          No
B          5          12         10         Yes
C          12         15         15         No
```

Rubric

Your grade will be based on the following scoring system:

Criteria	Score
Task Class	**5pts** for every correctly implemented method
Simulator Input	**0pts** Does not process the user input **5pts** Process user input but with errors **10pts** Correctly process all user input **15pts** Created a separate function to process user input correctly
Simulator Output	**0pts** No report **5pts** Has incorrect detail in the report **10pts** Correctly outputs the report **15pts** Implements the report generation correctly as a separate function