**COMP 482** **Project 2** **Fall 2023**

**Problem:** We have a set of jobs (each with a processing time and a deadline) that need to be scheduled on a single machine.

Write a Java program using:

a) **Earliest Deadline First** which sorts the jobs by deadline $d_i$ and schedules them in that order.

b) **Shortest Job First** which sorts the jobs by processing time $p_i$ and schedules them in that order.

c) **Least Slack First** which sorts the jobs by their slack $d_i - p_i$ and schedules them in that order.

**Input Format:** Create **input2.txt** file in the same directory as the java and class files. "**input2.txt**" will be entered as a command line argument. The first line will be n(the number of jobs). The next n lines will represent jobs, each with two values $p_i$ and $d_i$.

**Output Format:** Your program will output the number of jobs that each algorithm does not finish on time.

**Examples:**

If input1.txt contains:

```
4

2 2

1 3

4 7

3 4
```

Then the output will be:

Earliest Deadline First: 2 → It means {(2,2),(1,3),(3,4),(4,7)} has jobs (3,4) and (4,7) late.

Shortest Job First: 3 → It means {(1,3),(2,2),(3,4),(4,7)} has jobs (2,2) , (3,4) and (4,7) late.

Least Slack First: 3 → It means {(2,2),(3,4),(1,3),( (4,7)} has jobs (3,4), (1,3) and (4,7) late.

If Input1.txt contains:

4

3 5

7 10

1 11

5 16

Then the output will be:

Earliest Deadline First: 0 → It means {(3,5),(7,10),(1,11),(5,16)} has all jobs on time.

Shortest Job First: 1 → It means {(1,11),(3,5),(5,16),(7,10)} has job (7,10) late.

Least Slack First: 0 → It means {(3,5),(7,10),(1,11),(5,16)} has all jobs on time.

**Note:**

Your Java program should be **commented**, **indented**, and **structured**. Output should be sent to **System.out**. The program should be named project2. Please place all your files (.java, .class, .txt) in a directory named after you, zip them and submit them to canvas. Don't include any extra files and directories from IDE environment. The program must compile with the command **javac *.java** and run with the command **java project1 input2.txt**. Remember input2.txt is the command line argument.

Don't place the classes in a package (use default package).