You are given **n** jobs, each defined by a job identifier **JobId**, a start time s_i and an end time e_i, and **two machines: M1** and **M2**. A job can be scheduled on any machine, but each machine can only process one job at a time. The objectives are: (i) to design a **greedy algorithm** for job scheduling on these two machines in such a way that the **total number of non-overlapping jobs scheduled** is maximized, and (ii) to implement it in Python 3.X.

The input is given as a list of **n** jobs, where each job is defined by its JobID, start time and end time. The input must be represented as a sequence of triples **(JobID, s_i, e_i),** as shown in the sample input below.

Constraints:

1. Each job has a specific start time and end time.

2. A job should be run on only one machine.

3. Each machine can run only one job at a time.

4. The algorithm should attempt to fit jobs on the machines in a way that maximizes the number of jobs scheduled. If both machines are available for a job (that is, both are free), prioritize **M1** over **M2** to schedule.

5. All input times are positive integers.

6. Assume that $0 \le s_i < e_i \le 10^6$.

Sample Input (containing 6 jobs numbered 1 through 6):

```
6

1     2  8

2     3  5

3     1  4

4     6  10

5     7  9

6     5  7
```

The output must contain the total number of jobs that can be scheduled without overlap, along with the JobIDs of the jobs scheduled on **M1** followed by those scheduled on **M2**. Please follow the given output format strictly as any deviation will be penalized. (Spacing is arbitrary.)

Sample Output:

4

3 6

2 5

Additional Requirements:

> **python assignment2.py Input.txt**

The execution command line should take only one argument, which is the input file. The input file must be in the same directory as the program file, and the output file must be written in the same directory too. Furthermore, if the argument is **Input.txt**, then the output file must be named **Output.txt**. Similarly, for **Input0.txt, Input1.txt,** ... the output file must be name **Output0.txt, Output1.txt**,....Please follow these conventions strictly as any deviation will be penalized.

TURNIN: Upload, to Assignment 2 DropBox on Pilot, one zip archive per team, containing at least the following text files and a subfolder as specified below (with team member names and email addresses included in the first two files):
   (1) ReadMe.txt
   (2) Python 3.X source code file **assignment2.py** (containing algorithm pseudocode abstracting your implementation, greedy heuristic used and an informal argument for its correctness, and overall computational complexity, all succinctly included as documentation/comments)
   (3) Sample input and corresponding generated output (in the format shown above)
   (4) A separate subdirectory/subfolder containing other test inputs and outputs you used.

Include only one submission per team – not one submission per person. Please do not send redundant submissions by email. No late submissions are allowed. We will grade your final submission only so make a submission a day before to play it safe.