Vincent Nguyen
Term Project – Simulation

1. The data structures I used for D in this project is ArrayList. Firstly, because it is dynamic sizing, so I don't have to manually increase the size of the array and especially when I don't know the number of processes in advance. With using ArrayList I can make sure that the array can grow as needed to accommodate all the processes. Secondly, the run time for accessing the element in the ArrayList is O (1) by using the index.
2. When we want to execute the process which had equal priority and earlier arrival time instead of choosing arbitrarily. We can implement the modify the priority queue's comparator. The comparator should compare the processes based on two criteria which are priority and arrival time.
   - When I initiate the priorityQueue I will attach the custom comparator in it, so it will compare the priority and the arrival time of the processes

```
HeapAdaptablePriorityQueue<Integer, Process> priorityQueue = new
HeapAdaptablePriorityQueue<Integer, Process>(
   new Comparator<Process>() {
      @Override
      public int compare(Process p1, Process p2) {
         // Compare based on priority first
         int priorityComparison = Integer.compare(p1.getPriority(), p2.getPriority());

         if (priorityComparison == 0) {
            int arrivalTimeComparison = Integer.compare(p1.getArrivalTime(),
p2.getArrivalTime())
            return arrivalTimeComparison;
         }

         // If priority is different.
         return priorityComparison;
      }
   }
);
```

3. What other changes I could consider making to my project: I could break main method from my project to several methods or class instead put all under main method to reduce the complexity of the main method such as
   1. I could write the class for reading and writing the files with several methods where the input file is read, process objects are created, and write methods to write to the file when I call it.
   2. I could consider creating method to update wait times and update the priority.

   With those to changes it could improve my project efficiency

I could also consider looking to use a data structure that allows direct access to the element and could potentially optimize the process that traverse the entire priority queue to update the priories and wait times.