Name: Phuc Le

Course: CS146

Date: 10/20/17

Report Programming Assignment 1

1. Key concepts of this programming design.
2. Implement Max-Heap tree data structure in simulating a Priority Queue.
3. Use Priority Queue to illustrate the queue in the hospital.
4. Simulating normal activities in a hospital by using Priority Queue’s method.
5. Classes:
6. Patient.

* Purpose: Object the contain patient’s information help manage patient more effective.
* Attribute:
  + Name: contain name of the patient.
  + Condition: contain current condition of the patient. (The greater the number, the higher priority).
* Method:
  + getName: Return name of patient.
  + getCondition: Return current condition of patient.
  + setName: set the name of patient.
  + setCondition: set the condition of patient.
  + getInfo: return all information of a patient including name and condition.

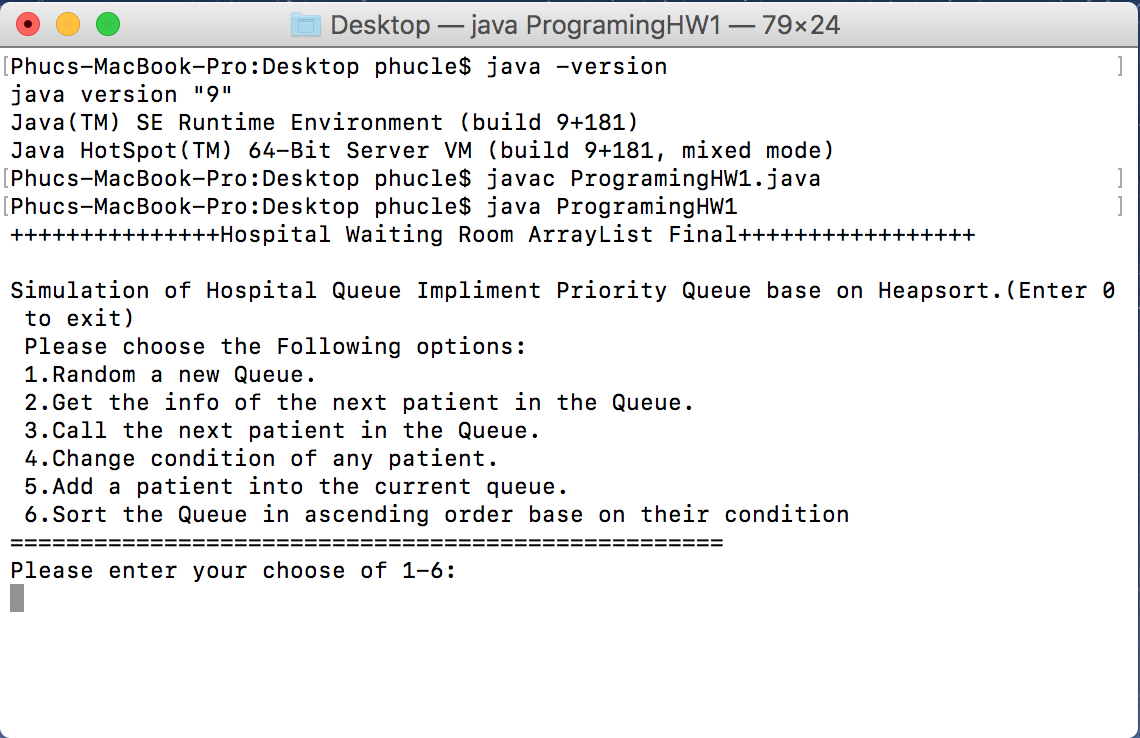
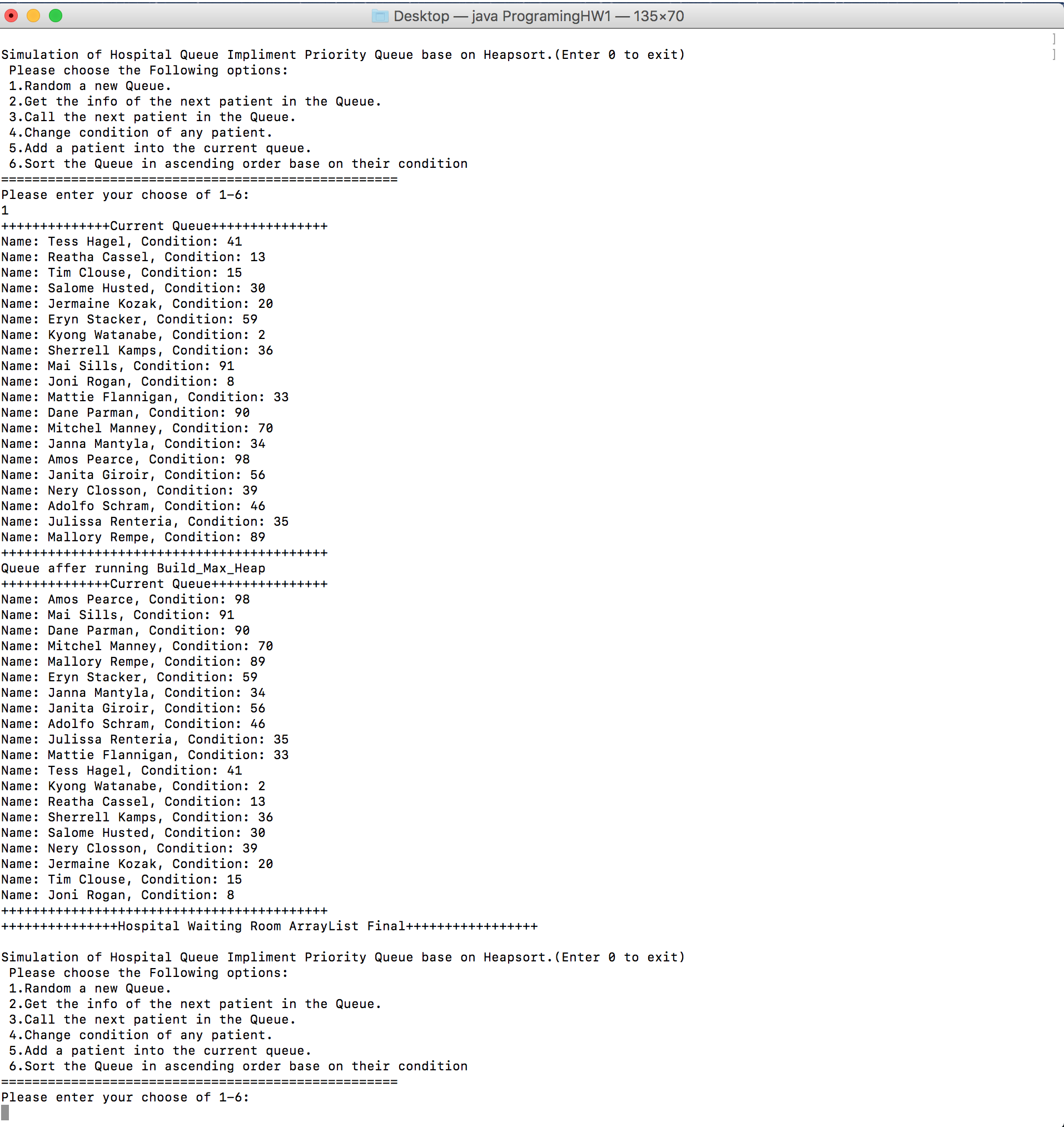
1. HeapsortArraylist.

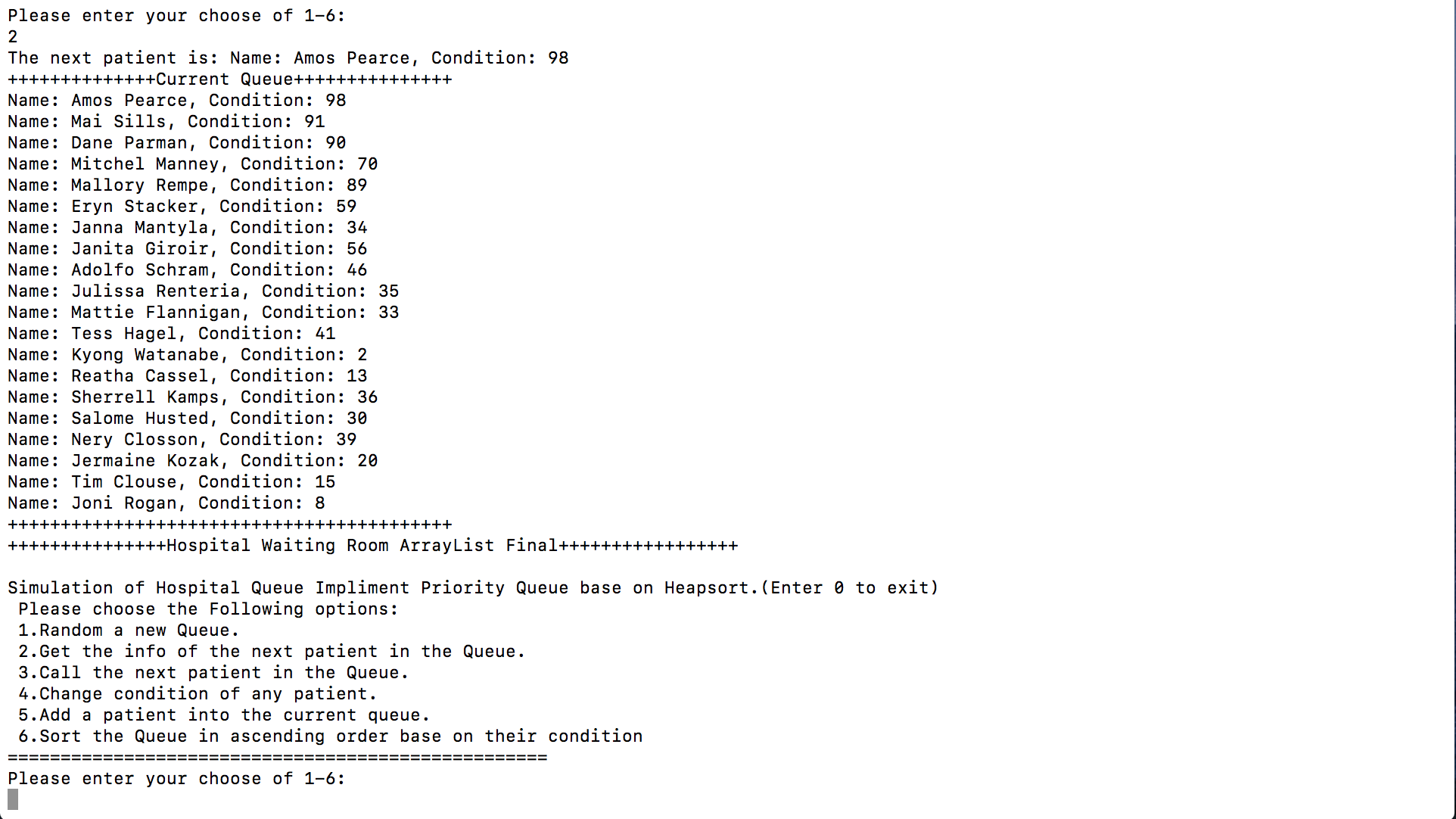
* Purpose: Heapsort class contains all the needed tools to implement Priority Queue.
* Attribute:
  + heap-size: variable that keep track of the current heap size.
  + largest: variable that used in the Max\_Heapify method.
* Method:
  + swaps: Used to swaps two elements in a array.
  + Max\_Heapify: Used for maintaining the max-heap property.
  + Build\_Max\_Heap: Used for building max-heap tree.
  + heapsort: Used for sorting the array.
  + Max\_Heap\_Insert: Used to insert new element into the heap tree but make sure to keep the max-heap property.
  + Heap\_Maximum: return the current highest condition.
  + Heap\_Increase\_Key: Used to increase any key( change condition of a patient).
  + Heap\_Extract\_Max: Used for getting and removing the current largest condition patient.

1. ProgrammingHW1.

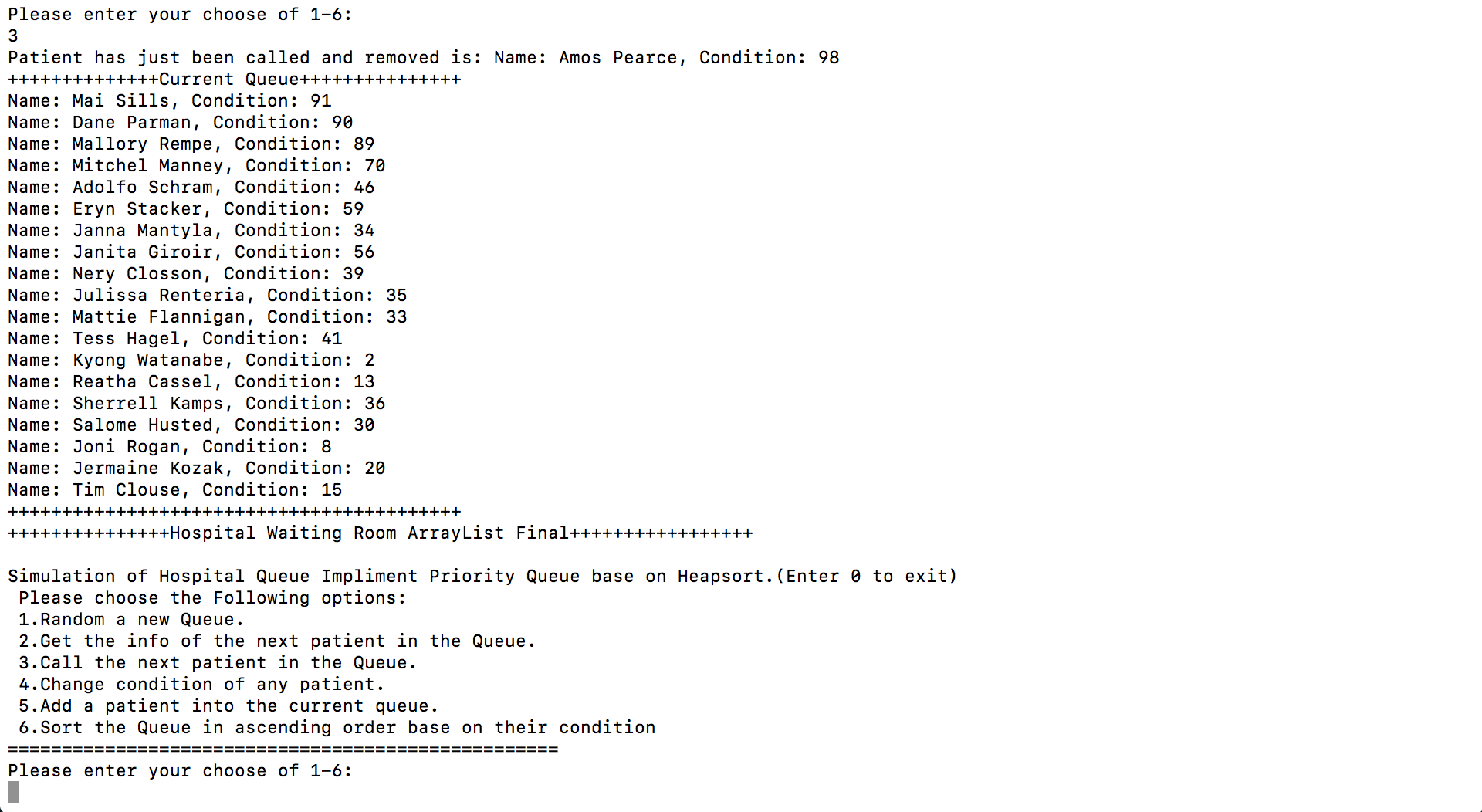
* Purpose: Main running class, which demonstrating the activities of a priority queue.
* Attribute:
  + name: array of random distinct names.
  + Patients: array of Object Patient.
  + Heapsort: instance of the HeapsortArraylist class.
  + currentUsedIndex: variable used to keep track of the usage of name in the name array. (Which make sure that we don’t have two patients with the same name.)
  + currentOccupiedSeat: Keep track of number of patients in the queue at the moment.
* Method:
  + RandomAPatient: Random a distinct patient with distinct name and condition.
  + RandomPatients: Random an array of distinct patients.
  + Print: Print out the console the current queue.
  + ValidateCurrentQueue: Check if the current queue has items and qualify the max-heap tree structure.

1. Screenshots and procedure to verify this code.

* After opening the file in terminal This window appear:
* Input “1”: Create a random Array and then print out, then Print out the queue after running Max\_Heapify: 
* Input “2”: Printout the patient that have the highest condition (most value in the Priority Queue):



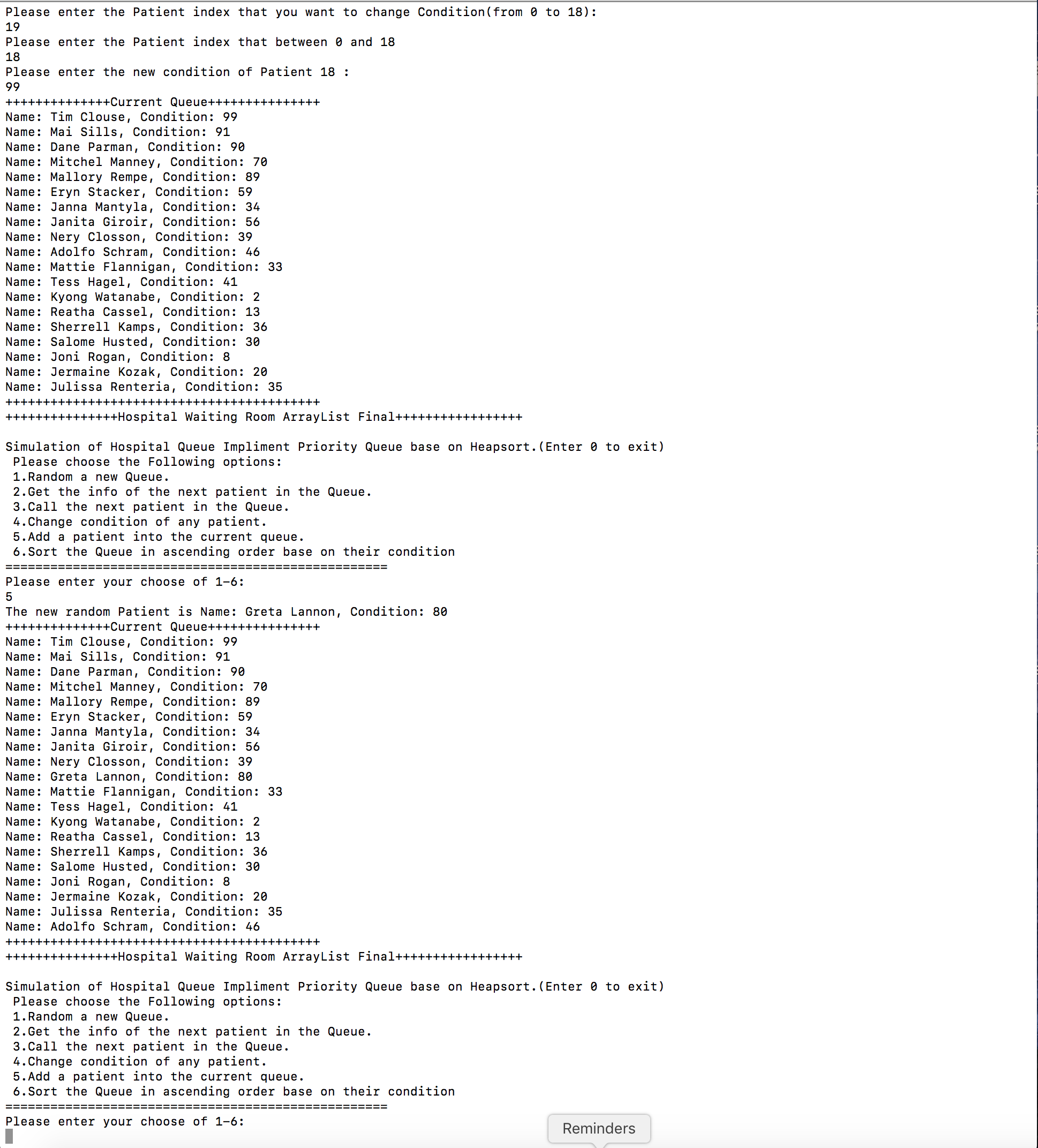
* Input ”3”: Remove and print out the patient that have the highest condition (most value in the Priority Queue):



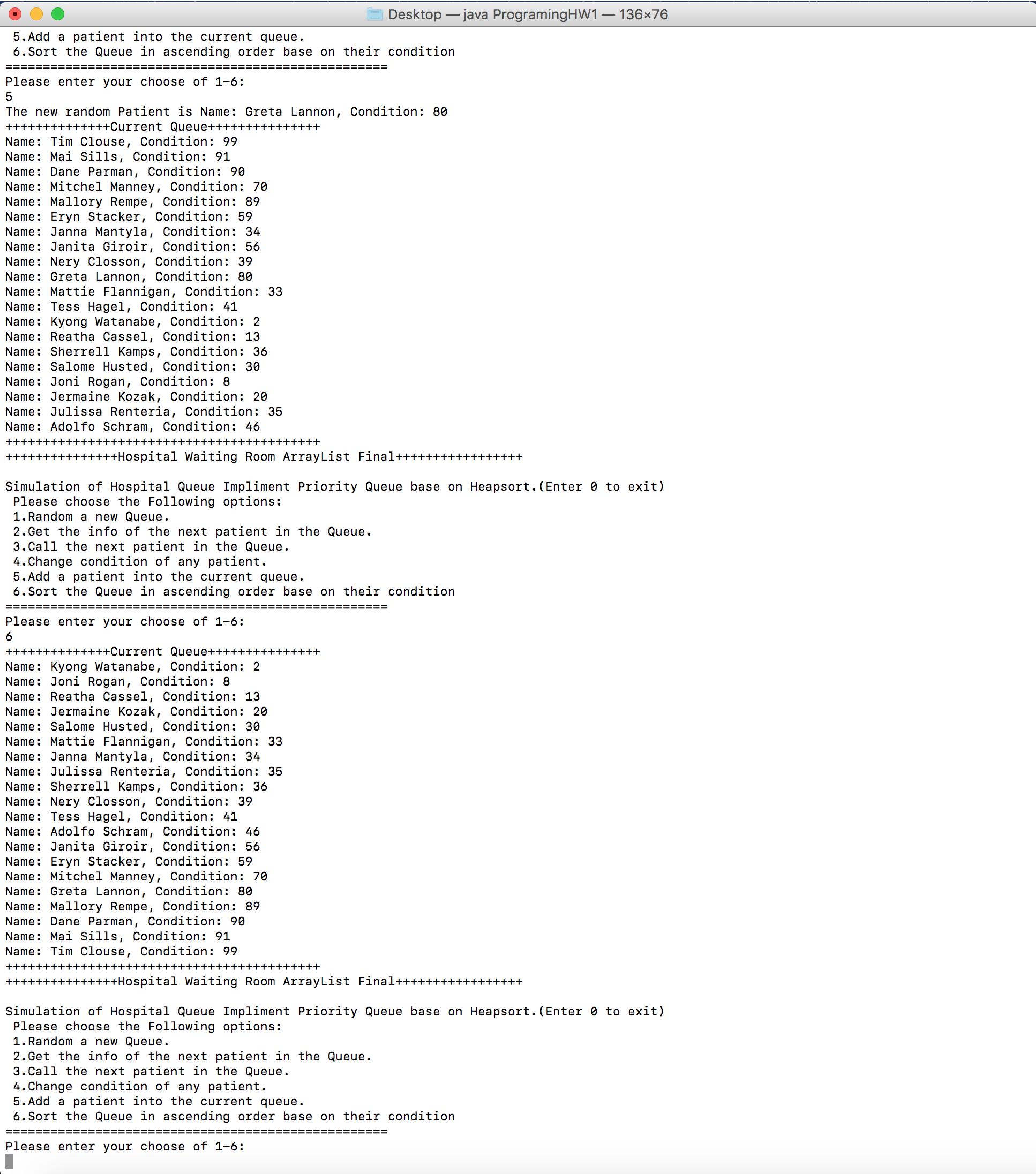
* Input “4”: Change condition of a patient:
  1. Input the Index of the patient that you wish to change condition, which have to be in range of 0-19.
  2. Then, Input the new condition of the patient, which has to be greater or equal to the current condition.



* Input “5”: Add a new random Patient into the queue, and printout the newly insert patient as well as the current Queue after insert. (The new queue will also an Max-heap tree).

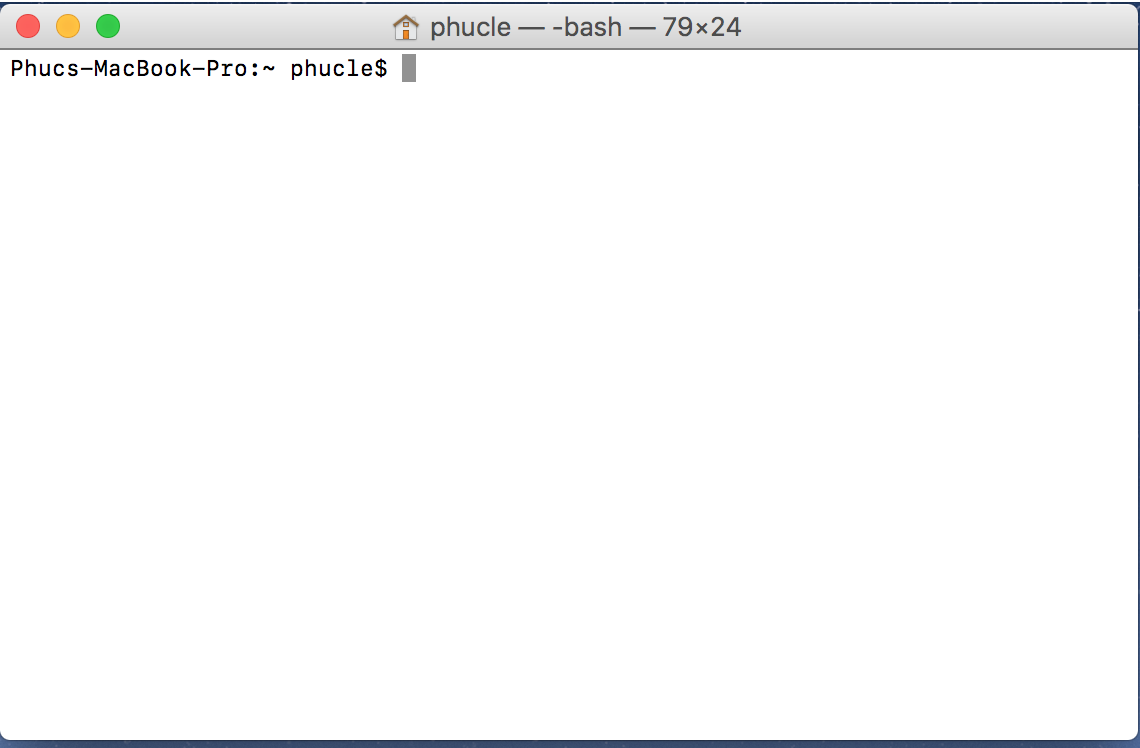


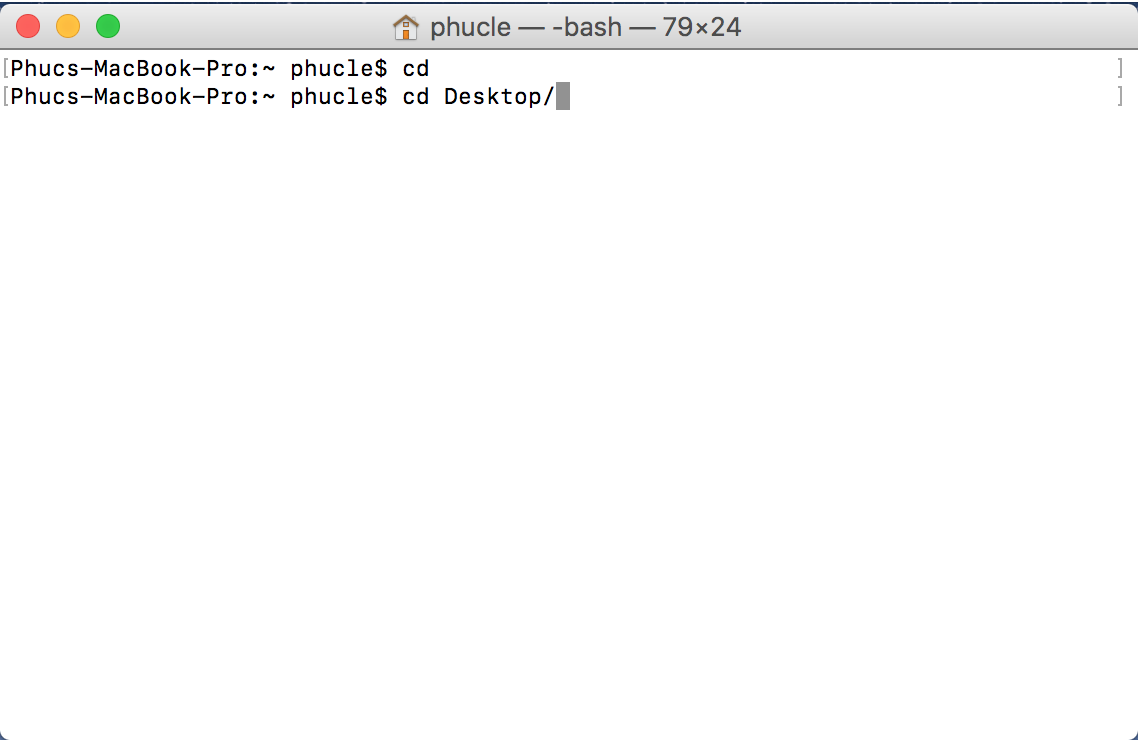
* Input “6”: Sort the current Array Using Heapsort and printout the array by ascending order base on their condition.

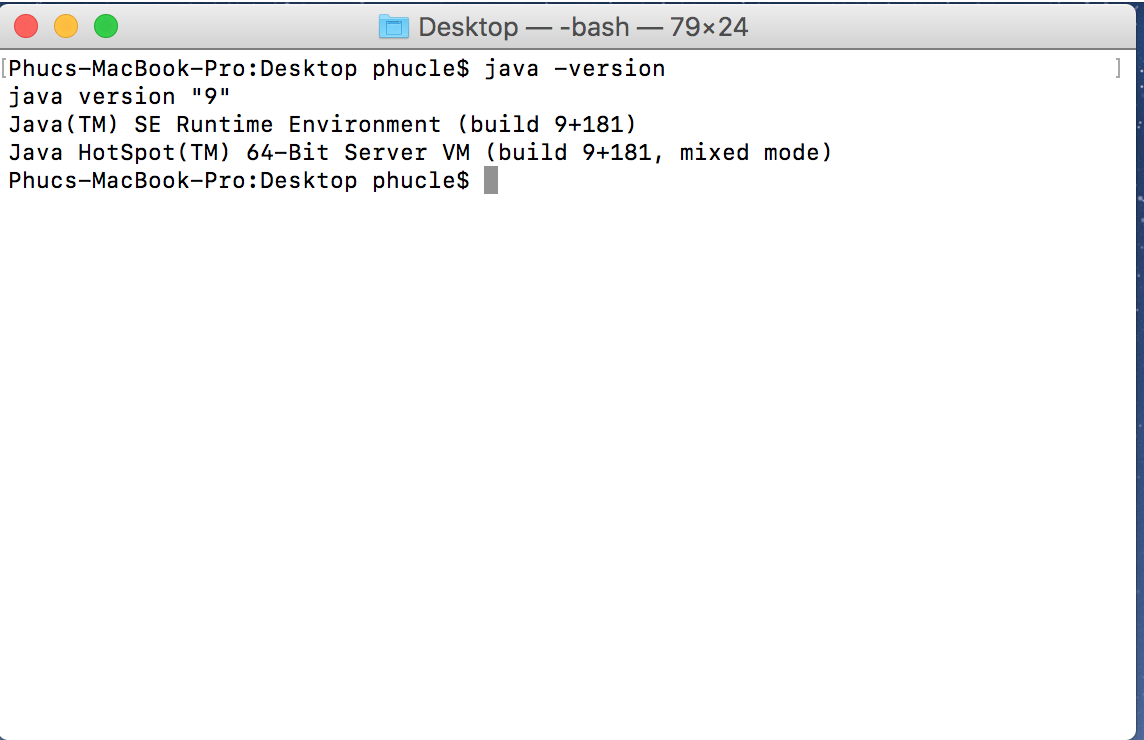


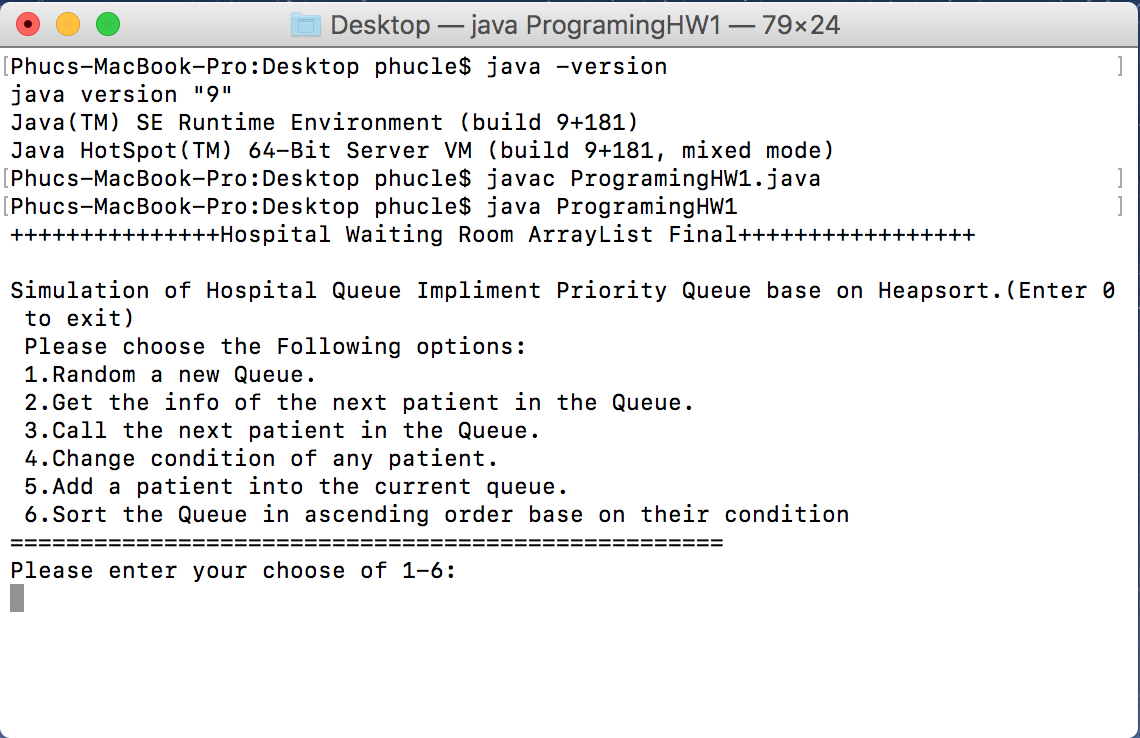
1. How to unzip and run the program:

* Download the zip file named Le-PA1.zip and place in on your desktop.
* Unzip the file will create a folder named Le-PA1.
* Copy and Paste the .java file named ProgrammingHW1.java on your desktop.
* Open your terminal (Or command line if you are using window).
* This window will appear:



* Type “cd” then enter.
* Type “cd Desktop/” then enter.
* Type “java –version” to check if your computer has installed the Java SE runtime environment. If you see the below screen you are Ok. If you are not seeing this screen please download the Java RE and JDK, which can be found here:

<http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>

* Type “javac ProgrammingHW1.java” and enter, then type “java ProgrammingHW1” and enter. The program should be running now an the following window will appear:

1. Problems encountered during the implementation.

* You might want to create another queue of patients (input = 1) after Heapsort(input = 6). Because after heapsort the array of patient is not a max-heap anymore.
* Insertion of a new patient is done automatically by random a new patient and add in the queue. So You might not able to add a new patient into the queue if you have not remove any patient since the queue is limited to 20.
* The random new patient will get a new distinct name and condition so if you try to add a lot of new patient, you will encounter problems because the array of distinct name and condition is limited.
* The input for changing condition of a patient must make sense for the program to get the best result.

1. Lesson Learned.

* The different of implementing ArrayList and array for Priority Queue.
* How to get an array with different values.
* Understanding heapsort and how Priority Queue work.