Name: Islam Osama Nwishy

ID#: 900170200

Assignment 2

1. **DEQ (Part 1)**
   * **Private:**
     1. **The Node [class]:** Contains a variable of whatever type declared by the user and 2 pointers one to point to the next node and one to point to the previous node.
     2. **Front and rear [pointers]:** 2 pointers, one that points to the most top node and one points to the most bottom node.
     3. **Size [variable]:** an integer that represents the size of the deque (Number of nodes).
   * **Public:**
     1. **DEQ [constructor]:** Instantiates front and rear pointers to NULL and size variable to 0.
     2. **DEQ [Deconstructor]:** deletes the front and rear pointers and sets the size to 0.
     3. **DEQisEmpty ():** returns true if the size is equal to zero.
     4. **Add\_front ():** adds a node at the top of the deque, makes it the new front, and makes the old front point to the new front as the previous node.
     5. **Add\_Rear ():** adds a node to the bottom of the deque, makes it the new rear and makes the old rear point to the new one as the next node.
     6. **Remove\_Front ():** removes the top node and returns its value.
     7. **Remove\_Rear ():** removes the rear node and returns its value.
     8. **View\_Front ():** returns the top node without removing it.
     9. **View\_Rear ():** returns the rear node without removing it.
     10. **DEQ\_Length ():** returns the size of the deque (returns the size variable).
2. **App (Part 2)**
   * **Plane [class]:** contains 2 private variables one for the number of the job the given plane is and the other is the time of arrival.
   * **Start ():** Declares andinstantiates the variables, then it runs the loop of the simulation where it increments time and calls the arrival and service functions.
   * **Arrival ():** generates a random number for the possibility of a plane arriving. If that random is bigger than the fixed probability of plane arrival, then it adds the plane to the queue, if it isn’t full, and logs in the job arrival time.
   * **Service ():** if the remaining time for service is 0 and the queue isn’t empty, it calls the function ExitLine () and starts the service based on a random value respective to the given average service time.
   * **ExitLine ():** it removes an element from the queue, calculates the wait time and logs in that wait time and start service time.
   * **AverageWait ():** calculate the average wait time of the simulation after it ends.
   * **Rand ():** will return a random float value between 2 numbers.
   * **Log ():** will output the log to the screen based on the value of the wait time. If the wait time is in negative it means that the user wants to log in an arrival time to the queue else, it means that the user wants to log in a landing time on the runway.
   * **QisFull ():** returns true if the queue length is more than or equal to the maximum size given by the user.