Bao Nguyen

Bcn140030

CS 1337.007

PROJECT 4 PSEUDOCODE

**Functions summary:**

Void read\_file(fstream &, object): read data from the input file to decide the time and date of the ships. Pass the file and the object by reference. Does not return anything.

Void write\_ticket(fstream &): write tickets to the output file. Pass the file by reference. Does not return anything.

Node \*enqueue(node \*&, int, int, int): pass the time and the head of the queue to the function. Add new node to the end of the queue and update the last node. Return last node.

Void dequeue(node \*&, int, int, int): pass the time and the head of the queue to the function. When the ship leaves the station, remove it from the beginning of the queue. Does not return anything.

Bool check\_available(int, int, int): pass the time in to check the availability. Return true if available, else otherwise.

Int minutes\_over(int, int, int): pass the time in to calculate the minutes the ship stays over the reserved time. Return the number of minutes over.

Float calculate\_fine(int, float): calculate the fines by multiplying the minutes over by the fine rate. Return fine.

Void get\_data(object): get data (ship name, ID, time in, time out, etc.) from the user. Does not return anything.

Bool check\_violation(int, int, int): pass the time in to check if the ship stays overtime. Return true if it does, else otherwise.

**Logical flow:**

Read\_file

Pass the input file and the object by reference

While (not end of file & dock <= 10)

Skip everything until the first number

Put first integer to hour

Skip the semicolon

Put the second integer to minute

Skip the space

Put in ship name

Skip the space

Put in ship ID

Skip the space

Put in the integer for paid time

Main

Call function read\_file and store in the array of 10 elements

Call function get\_data

Call function check\_time

If time is not available

Call function enqueue

Else

Put the ship in to the array

When the time is up

Call function dequeue

If violation time

Call function write ticket

Get data

Ask user for the following info: ship name, ship ID, minutes

Validate every input

Write ticket

If the ship stays overtime, write its ticket to the file called “tickets.txt” by creating the new file stream

Write data with the designed file template

Pass the template into the overloaded operator of the ticket class to write to the file

Enqueue

Pass in the time in

Add the new node containing the time to the end of the list

Update the last node of the list

Dequeue

Pass in the time out

Remove nodes from the beginning of the list when the ships leaves

Check violation

Calculate minutes from input

If (hour\_in = hour\_out)

Then actual\_time = minute\_out – minute\_in

Else

If (minute\_in <= minute\_out)

Then actual\_time = (hour\_out – hour\_in)\*60 + (minute\_out – minute\_in)

Else

Then actual\_time = (hour\_out – hour\_in - 1)\*60 + (60 – minute\_in + minute\_out)

Now compare times

If (actual\_time > reserved time)

Return true;

Else

Return false;

Check\_available

Pass in array

Elements = sizeof(array) / sizeof(each element)

If (elements < 10)

Return true

Else

Return false