Project 2: Priority of Passengers

Fall 2019

Due: Saturday, November 23, before 11:59 PM

Abstract

An airline company uses the formula shown below to determine the priority of passengers on the waiting list for overbooked flights.

Priority Number = (A / 1000) + B - C

where

A is the customer's total mileage in the past year

B is the number of years in his or her frequent flier program

C is a sequence number representing the customer's arrival position when he or she booked the flight.

Given a file of overbooked customers as shown in the table below, write a program that reads the file and determines each customer's priority number. The program then builds a priority queue using the priority number and prints a list of waiting customers in priority sequence.

Name	Mileage	Years	Sequence
Rose King	93,000	2	1
Dolores Ramirez	17,000	5	2
Evelyn Bush	42,000	5	3
Jerry Dawson	48,000	3	4
Phil Kim	69,000	7	5
Jose Mills	26,000	1	6
Irving Gregory	41,000	3	7
Adrian Berry	85,000	4	8
Nicholas Owen	76,000	6	9
Betsy Joseph	31,000	7	10
Margie Wilkins	38,000	2	11
Tammy Powell	62,000	1	12
Susan Johnson	47,000	3	13
Chris Foster	18,000	6	14
David Phillips	92,000	4	15

Attachment

maxBinaryHeap.c

customers.txt

Deliverables

Project submissions should include a program file along with screenshots of your output. Your file name should be “Project-2-<LastName>.c”

Output

```
Name: Rose King -> Priority: 94
Name: Adrian Berry -> Priority: 81
Name: David Phillips -> Priority: 78
Name: Nicholas Owen -> Priority: 73
Name: Phil Kim -> Priority: 71
Name: Tammy Powell -> Priority: 51
Name: Jerry Dawson -> Priority: 47
Name: Evelyn Bush -> Priority: 44
Name: Irving Gregory -> Priority: 42
Name: Susan Johnson -> Priority: 37
Name: Margie Wilkins -> Priority: 29
Name: Betsy Joseph -> Priority: 28
Name: Jose Mills -> Priority: 21
Name: Dolores Ramirez -> Priority: 20
Name: Chris Foster -> Priority: 10
```

Rubric

TOTAL 100 Points

- 40 Points – Code compiles and runs
- 10 Points – Correctly reads the text file line by line
- 10 Points – Correctly breaks each line into a series of tokens (**Name, Mileage, Years, Sequence**) using the delimiter **(tab → ‘\t’)**
- 10 Points – Correctly calculates the priorities
- 20 Points – Correctly prints the name and priority number of customers
- 10 Points – Code is readable and well documented (comments, proper white spacing, good variable/function names)