Due Wed Oct 17 at the start of your lab section; Submit
Server: class = cse2010, assignment = hw4SzIndividual
Due Wed Oct 17 at the end of your lab section; Submit
Server: class = cse2010, assignment = hw4SzGroupHelp
x is 2, 3, or 4—your section number (or “j” for java).

To improve the experience of customers and potentially rev-

To implement the priority queue, you may modify/rewrite Pro-

Input: Input is from the command-line arguments for hw4.c
in this order:

Output: Output goes to the standard output (screen), each
line corresponds to an action:

Extra Credit (10 more points) For the priority queue,
use bottom-up heap construction (heapifying a complete bi-
ary tree) to efficiently initialize the priority queue from data
in the first input file. Add function/method initialize(pQueue,
entryArray, numEntries) near the top of one of your program
files.

Substitution: Submit hw4.c (and hw4extra.c for extra
credit) that has the main method and other program files. Sub-
missions for Individual and GroupHelp have the same guide-
lines as HW1. Note the late penalty on the syllabus if you
submit after the due date and time as specified at the top of
the assignment. Late submission is NOT available for extra
credit.

<table>
<thead>
<tr>
<th>Functions/methods include:</th>
<th>insert(pQueue, entry)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>removeMin(pQueue)</td>
</tr>
<tr>
<td></td>
<td>getMin(pQueue)</td>
</tr>
<tr>
<td></td>
<td>isFull(pQueue)</td>
</tr>
<tr>
<td></td>
<td>isEmpty(pQueue)</td>
</tr>
</tbody>
</table>

To implement the priority queue, you may modify/rewrite Pro-

Input: Input is from the command-line arguments for hw4.c
in this order:

1. Filename of initial song ratings. The first line has the
target customer name. Each of the following lines has the
name of a customer and 10 song ratings. Each rating is
between 1 (poor) and 5 (excellent), or 0 (not rated).

2. Filename of actions, each line has one of the following actions:
   - AddCustomer custName rating0 rating1 ... rating9
   - RecommendSongs
   - PrintCustomerDistanceRatings

Output: Output goes to the standard output (screen), each
line corresponds to an action:
   - AddCustomer custName rating0 rating1 ... rating9
   - RecommendSongs closestCustName/none song1 rating1 song2 rating2 ...
   - PrintCustomerDistanceRatings

PrintCustomerDistanceRatings prints the target customer first, then
the other customers in alphabetical order in a table. Distance
starts at column 1, customer starts at column 7, rating0 is at
column 18, rating1 is at column 19, ... Print distance with
three decimal places (x.xxx), for example, 1.234 or ----- if the
distance cannot be calculated.

Sample input files and output are on the course website.

Extra Credit (10 more points) For the priority queue,
use bottom-up heap construction (heapifying a complete bi-
ary tree) to efficiently initialize the priority queue from data
in the first input file. Add function/method initialize(pQueue,
entryArray, numEntries) near the top of one of your program
files.

Substitution: Submit hw4.c (and hw4extra.c for extra
credit) that has the main method and other program files. Sub-
missions for Individual and GroupHelp have the same guide-
lines as HW1. Note the late penalty on the syllabus if you
submit after the due date and time as specified at the top of
the assignment. Late submission is NOT available for extra
credit.

<table>
<thead>
<tr>
<th>Functions/methods include:</th>
<th>insert(pQueue, entry)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>removeMin(pQueue)</td>
</tr>
<tr>
<td></td>
<td>getMin(pQueue)</td>
</tr>
<tr>
<td></td>
<td>isFull(pQueue)</td>
</tr>
<tr>
<td></td>
<td>isEmpty(pQueue)</td>
</tr>
</tbody>
</table>

To implement the priority queue, you may modify/rewrite Pro-

Input: Input is from the command-line arguments for hw4.c
in this order:

1. Filename of initial song ratings. The first line has the
target customer name. Each of the following lines has the
name of a customer and 10 song ratings. Each rating is
between 1 (poor) and 5 (excellent), or 0 (not rated).

2. Filename of actions, each line has one of the following actions:
   - AddCustomer custName rating0 rating1 ... rating9
   - RecommendSongs
   - PrintCustomerDistanceRatings

Output: Output goes to the standard output (screen), each
line corresponds to an action:
   - AddCustomer custName rating0 rating1 ... rating9
   - RecommendSongs closestCustName/none song1 rating1 song2 rating2 ...
   - PrintCustomerDistanceRatings

PrintCustomerDistanceRatings prints the target customer first, then
the other customers in alphabetical order in a table. Distance
starts at column 1, customer starts at column 7, rating0 is at
column 18, rating1 is at column 20, ... Print distance with
three decimal places (x.xxx), for example, 1.234 or ----- if the
distance cannot be calculated.

Sample input files and output are on the course website.

Extra Credit (10 more points) For the priority queue,
use bottom-up heap construction (heapifying a complete bi-
ary tree) to efficiently initialize the priority queue from data
in the first input file. Add function/method initialize(pQueue,
entryArray, numEntries) near the top of one of your program
files.

Substitution: Submit hw4.c (and hw4extra.c for extra
credit) that has the main method and other program files. Sub-
missions for Individual and GroupHelp have the same guide-
lines as HW1. Note the late penalty on the syllabus if you
submit after the due date and time as specified at the top of
the assignment. Late submission is NOT available for extra
credit.

<table>
<thead>
<tr>
<th>Functions/methods include:</th>
<th>insert(pQueue, entry)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>removeMin(pQueue)</td>
</tr>
<tr>
<td></td>
<td>getMin(pQueue)</td>
</tr>
<tr>
<td></td>
<td>isFull(pQueue)</td>
</tr>
<tr>
<td></td>
<td>isEmpty(pQueue)</td>
</tr>
</tbody>
</table>