To add an element of surprise/excitement (and perhaps more revenue), an online auction site tries to sell multiple items of the same product at random times over a period of time (e.g. 1000 items at random times over 24 hours). How would you design an efficient system to match items to bids?

The goal of HW4 is to design a system that can efficiently match items with the highest bidding price at different times. The system allows customers to enter, remove, and change bids. Each bid consists of a price and quantity. For simplicity, each customer can have one bid and there is only one product. If two bids have the same price, the earlier bid has a higher priority (assume the time of a bid is unique). Also, to not lose money, the auction site does not sell an item if the highest bidding price is lower than the minimum acceptable price, which can be updated over time by the retailer (e.g. higher at the beginning).

To manage and find the highest bidding price efficiently, use HeapAdaptablePriorityQueue (textbook). You may modify HeapAdaptablePriorityQueue and related classes to increase their general functionality. The program files are on the course website.

Input: The command-line argument for HW4.java is the name of the input file, which has:

- EnterBid time name price quantity
- RemoveBid time name
- ChangeBid time name price quantity
- UpdateMinimumAcceptablePrice time price
- SellOneItem time
- DisplayHighestBid time

Time is an integer in HHMM format, where HH is 00-23 and MM is 00-59 (leading zeros are optional). Sample input files are on the course website.

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

- EnterBid time name price quantity [ExistingCustomerError]
- RemoveBid time name [noCustomerError]
- ChangeBid time name price quantity [NoCustomerError]
- UpdateMinimumAcceptablePrice time price
- SellOneItem time name price [NoBids/HighestBiddingPriceIsTooLow]
- DisplayHighestBid time name price quantity

Sample output is on the course website.

Submission: Submit HW4.java that has the main method, (modified) HeapAdaptablePriorityQueue.java and related classes, and other program files. Submissions for Individual and GroupHelp have the same guidelines 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.