University of Waterloo
CS240 Fall 2020
Programming Question 1

Due Date: Wednesday, September 30 at 5:00pm

The integrity of the grade you receive in this course is very important to you and the University of Waterloo. As part of every asessment in this course you must read and sign an Academic Integrity Declaration before you start working on the assessment and submit it **before the deadline of Sept 30** along with your answers to the assignment; i.e. **read, sign and submit PQ01-AID.txt now or as soon as possible**. The agreement will indicate what you must do to ensure the integrity of your grade. If you are having difficulties with the assignment, course staff are there to help (provided it isn't last minute).

The Academic Integrity Declaration must be signed and submitted on time or the assessment will not be marked.

Please read `http://www.student.cs.uwaterloo.ca/~cs240/f20/guidelines/guidelines.pdf` for guidelines on submission. Submit the file `teampq.cpp` to MarkUs.

Problem 1   [10 marks]

```
class Team {
   int wins;
   int losses;
   string name;
   // You may add fields/methods/constructors/destructor as necessary
};
```

We wish to build a data structure that allows for efficient retrieval of the team with the most wins and also for efficient retrieval of the team with the fewest losses. Essentally, we want a priority queue that supports two different priority measures. Provide an implemenation of the following class:

```
class TeamPQ {
   // add fields/methods/constructors/destructor as necessary
 public:
   void insert(const Team &t);   // O(log n) time

   const Team &findMaxWins() const;    // O(1) time
   const Team &findMinLosses() const;  // O(1) time
```

```
    void removeMaxWins();      // O(log n) time
    void removeMinLosses();    // O(log n) time
    void removeAllK(int k);     // ??? time
};
```

Implement this class in C++ and provide a main function that accepts the following commands from stdin (you may assume that all inputs are valid):

- `i wins losses name` - inserts a team with the given wins, losses, and name into the priority queue. You may assume that `name` contains no whitespace.

- `pw` - prints the name of the team with the most wins, without removing it from the priority queue. Prints nothing if the priority queue is empty.

- `pl` - prints the name of the team with the fewest losses, without removing it from the priority queue. Prints nothing if the priority queue is empty.

- `rw` - removes the team with the most wins from the priority queue and prints nothing. Does nothing if the priority queue is empty.

- `rl` - removes the team with the fewest losses from the priority queue and prints nothing. Does nothing if the priority queue is empty.

- `ra k` - removes all teams with k wins from the priority queue and prints nothing. Does nothing if the priority queue is empty. Removes no teams if no team has k wins.

Determine the runtime of `removeAllK` and write it at the top of the file in comments. Briefly justify (in a sentence or two) the runtime.

All output is printed to stdout. The program terminates when eof is encountered.

For example, the following input:

```
i 15 5 leafs
i 13 3 canucks
i 12 7 canadiens
i 10 10 jets
i 0 14 oilers
pw
pl
rw
pw
rl
pl
i 20 1 leafs
pw
pl
```

should produce the following output:

```
leafs
canucks
canucks
canadiens
leafs
leafs
```

You may not use any pre-existing code that would trivialize the implementation (e.g. heaps from STL, smart pointers, etc). You may, however, use the C++ vector data structure. If in doubt, make a private Piazza post and ask.

Place your entire program in the file `teampq.cpp`