Prelab : Week of April 13th

For this prelab you are to implement a priority queue ADT with the following interface functions:

```
/* The parameter is an error code (the values of
   which must of course always be documented). */
PQueue initPQueue(int * eCode)

/* This function inserts obj with its associated
   priority and returns an error code. */
int insertPQ(void* obj, double priority, PQueue q)

/* This function removes and returns the object
   with minimum priority in O(1) time. */
void* minDeletePQ(PQueue q)

/* This function removes and returns the object
   with maximum priority in O(1) time. */
void* maxDeletePQ(PQueue q)

/* This function returns the smallest priority
   value in the PQ in O(1) time */
double getMinPriority(PQueue q)

/* This function returns the largest priority
   value in the PQ in O(1) time */
double getMaxPriority(PQueue q)

/* This function returns the number of objects
   in the priority queue in O(1) time. */
int getSizePQ(PQueue q)

/* This function frees all memory associated with
   with the priority queue. Documentation should
   explicitly inform the user to not attempt to
   use the PQ after calling this function. */
void freePQ(PQueue q)
```

The only other implementation constraint is that your underlying data structure (simple linked list, doubly linked list, circular doubly linked list, or whatever) must make use of at least one dummy node to simplify the logic of your functions. NOTE: Your documentation should address any potentially undefined cases, e.g., attempting to delete from an empty queue (but that's not the only one!).

BONUS "FUN" STUFF: You might want to read about and include the following:

```
#include <float.h>
```

This gives access to the largest and smallest possible double values, e.g., `DBL_MAX` is the largest positive double value and `-DBL_MAX` is the most negative (smallest) possible double. It turns out this can be used to further simplify the logic of your insert function. How does it help? Well, here's a hint: Maybe there's a way to make use of the priority member of your dummy node(s).

- JKU