

## CX 4010 / CSE 6010 Assignment 1C: Intro to C (Part 3)

### Due Dates:

- Due: 11 AM, Friday, September 14, 2018

The objectives of this assignment are to write a program to implement a simple priority queue using a sorted, singularly linear linked list data structure. The priority queue code should be structured as if it were part of a software library to be used with other programs written by someone else. Small keys indicate higher priority. The software should implement the following interface:

1. Use a `typedef` to define a data structure called `PrioQ`, defined as a C `struct` representing the priority queue. The implementation of the data structure is not visible outside the library you are creating.

2. Write a function with the prototype

```
PrioQ *PQ_create();
```

Create a new priority queue and return a pointer to it. The priority queue is initially empty, i.e., contains no elements. Return `NULL` if the operation failed, e.g., due to a lack of memory.

3. Write a function with the prototype

```
int PQ_insert(PrioQ *PQ, double key, void *data);
```

Insert the item pointed to by `data` into the priority queue `PQ`. The priority of this data item is `key`. Return `NULL` if the operation failed, or an arbitrary value not equal to `NULL` if it succeeded. The data type of the inserted item pointed to by `data` is defined by the caller, and is not visible to the priority queue library.

4. Write a function with the prototype

```
void *PQ_delete(PrioQ *PQ, double *key);
```

Remove the data item from the priority queue `PQ` with the smallest key (priority). The function returns a pointer to the data item that was removed, or `NULL` if the queue was empty. The priority of the deleted item is returned in `key`.

5. Write a function with the prototype

```
Unsigned int PQ_count(PrioQ *PQ);
```

Returns the number of items currently residing in the priority queue.

6. Write a function with the prototype

```
void *PQ_free(struct PrioQ *PQ);
```

Delete the priority queue PQ by releasing all memory used by the contents of the data structure.

7. Write a program to test your priority queue library. The test program should first create a priority queue, create 20 data elements, assign each a priority drawn from a random number generator and insert it into the queue. Then remove the elements one after another and verify that the priority of the removed items is in increasing order. Include print statements to demonstrate correct operation.

Your code should include a .h file that defines the interface to the software that includes only the information necessary to use the library. It should *not* specify the internal implementation of the code.

Turn in your software in a single zip file. Your software must be well documented and include comments so the code is easy to understand. Also turn in a brief report describing how you tested your software, and verified each of the functions works correctly.

### **Grading**

Your grade will be based on the structuring and documentation of your code, and the brief report verifying the code works correctly. For this assignment you need not verify that your code runs on the deepthought cluster.

### **Collaboration, Citing, and Honor Code**

As a reminder, please refer to the course syllabus regarding rules and expectations regarding collaborating with other students and use of other resources, including materials available on the web.