

In-lab assignment for week 11

We'll assume the following structures:

```
typedef struct nodestruct {
    int key;
    float priority;
    struct nodestruct *next;
} Node;
typedef struct {
    int size;
    Node *front, *rear;
} PQueue;
```

In this lab, you need to implement six functions for a priority queue based on the given structure definitions above. Note: You could implement the functions in any way you choose and TAs will grade all your functions using a uniform testing program. However, you should follow the given structure declarations and function prototypes, and satisfy the required complexity. Changing declarations or prototypes will lead to a zero grade for this assignment.

`PQueue* initPQ();` /* this function creates an empty queue and returns it. This function receives no input, and returns a pointer. $O(1)$ complexity*/

`int insertPQ(int, float, PQueue*);` /* This function receives an integer key value, a float priority value, and the current priority queue pointer. It creates a new node with the key and priority values, inserts it into the priority queue **based on the priority value in ascending order**, then returns an error code: 0 if the insertion was successful, -1 otherwise. $O(n)$ complexity*/

`int deleteMinPQ(PQueue*, int*);` /* This function receives the current priority queue and an integer pointer as inputs. It removed the node with the **minimum** priority value, and stores the removed key value into the integer pointer. It returns an integer error code: 0 if the deletion was successful, -1 otherwise. $O(1)$ complexity.*/

`void reInitPQ(PQueue*);` /* This function receives the current priority queue, reinitializes the priority queue to an empty priority queue without changing the address of the priority queue. $O(n)$ complexity*/

`void freePQ(PQueue*);` /* This function receives the current priority queue and frees all the memories that have been previously allocated. $O(n)$ complexity.*/

`void printPQ(PQueue*);` /* This function receives the current priority queue, then prints out the key values on each node together with their priority values. For example:

Current PQ is: key 6 with priority 0.85 --> key 4 with priority 4.57 --> NULL

$O(n)$ complexity.*/

The assignment you need to submit should include everything **but** your main program, for example, your submission should look like this:

includes
Declarations of structures
Function prototypes
Function implementations

Grading Criteria:

initPQ function: 5 points

insertPQ function: 5 points

deleteMinPQ function: 5 points

reInitPQ function: 5 points

freePQ function: 5 points

printPQ function: 5 points

no partial credits will be given for incomplete/incorrect functions.

General note:

1. Command to compile your code in cmd window: `gcc labx.c -Wall -Werror`
2. If your code does not compile with “-Wall -Werror” flag, you will receive an automatic 0 for this assignment.
3. Changing the given function prototype or struct definition will lead to an automatic zero grade.
4. Using any global variables will lead to an automatic zero grade.
5. The implementation of the function should include comments describing what it is intended to do and how this function should be called. Example can be found in CS 2050 lab policy.
6. If your submission does not include a source file, you will receive an automatic zero grade.