In-lab assignment for week 12

We'll assume the following structure:

```
typedef struct {
   int arraySize; // the size of the sorted array
   int availableIdx; // next available index in the sorted array
   int *sortedArray; // the sorted array
} PQueue;
```

In this lab, you need to implement six functions based on the given structure definition 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(int); /* This function initializes a PQueue struct with availableIdx = 0, arraySize = the input integer, and creates an integer array with length = the input integer. It returns the initialized PQueue pointer.*/
int insertPQ(PQueue*, int); /* this function receives the current priority queue and a new integer value, then inserts the new integer into the sorted array in ascending order. It returns an error code: 0 if the insertion was successful, -1 if the insertion failed*/
int DeQueue(PQueue*, int*); /* this function receives the current priority queue and an integer pointer. It removes the maximum integer from the sorted array, stores the removed value into the integer pointer, then returns an error code: 0 if the deletion was successful, -1 if the deletion failed. This function has O(1) time complexity.*/
int findIdx(PQueue*, int); /* this function receives the current priority queue and a query integer for searching. Then it searches this query integer in the sorted array and returns the index of the query integer if it exists, or -1 if it doesn't exist. This function has O(log(n)) time complexity.*/
void freePQ(PQueue*); /* this function frees all the allocated memories.*/
void printPQ(PQueue*); /*this function prints the current sorted array.*/

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: 10 points

DeQueue function: 5 points

findIdx function: 10 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.