Assignment 1

Implementing an ADT for Priority Queue

Due: August 2, 2006

In this assignment, you will implement an abstract data type called **PQUEUE** that implements a priority queue that can store an **unlimited** number of integers. The data type PQUEUE should support the following operations (**with exactly the same function prototypes as given below**):

1. ***PQUEUE Init(PQUEUE Q)*** – initializes *Q* to an empty priority queue and returns the initialized priority queue.
2. ***int IsEmpty(PQUEUE Q)*** – returns 1 if *Q* is empty, 0 otherwise.
3. ***int Insert(PQUEUE Q, int x)*** – Inserts the value *x* in *Q*. Returns 0 if the insert is successful, -1 if there is any error in inserting.
4. ***int ExtractMax(PQUEUE Q, int *x)*** – Returns the maximum element in *Q* in the variable pointed to by *x*. *x* must point to an already allocated space. Returns 0 if successful, -1 if *Q* is empty, and –2 if there is any other error.
5. ***void Destroy(PQUEUE Q)*** – frees any allocated memory for *Q*. If no memory is allocated for *Q*, does nothing.

Design appropriate data structures to define PQUEUE and implement the above function. You can assume that the user will not call *Init( )* twice on the same queue and that all other functions are called only on an initialized queue. Your final output will be two files:

1. A .h file containing the type definition for PQUEUE and any other type definition you may need. Name the file <your roll no.>.h (for ex. 06CS1004.h)
2. A .c file containing the implementation of the above functions that can be compiled into a static library. For information on how to create a static library, look up the linux manpage for the "ar" command. You do not need to actually submit the static library, we will create it while evaluating. Name the file <your roll no.>.c (for ex., 06CS1004.c).

We will test your program by creating another C file that has a main function to call the above functions (including error condition checks). This C program will be linked with your static library while compiled. You should think about what all needs to be tested in your program and write a main function to test it appropriately.

It is very important that you follow the above file naming conventions and function prototypes EXACTLY as the evaluation will be done by a program that will assume these. Any error arising out of deviations from above will incur severe penalty in marks.