**Assignment 5: Implementing a templatized array-based max Heap ADT (HeapADT).**

**This can be done in pairs**

Part A: The ADT is called **HeapADT**

It has exactly two integer variables: count and size.

Required Functions:

1. void insert (T a)
2. T remove ()
3. HeapADT() //default constructor. Creates a heap array HA of default size = 50
4. HeapADT(int size) constructor. Creates a heap array of size = size

You will also need a destructor.

PartB: Develop an appropriate main/driver program to test the ADT.

Part C: Application of HeapADT: **Simulating priority queues**.

1. Create a program structure. It contains:

* float ProgramAge (representing creationTime),
* integer programNumber and
* integer priority\_level (0 = LOW, 1 = MID and 2= HIGH)**.**

ProgramNumber is a just like a counter starting from 0. Program age can have values 0<ProgramAge<15

1. Create three heaps of type program: HeapADT<program> low, high and mid.
2. Create a function **program getNextProgram()**, that creates a program, randomly assigns values to its programAge and priority\_level (0,1 or 2) and returns the program. It also assigns value to programNumber. let programNumber = 0 for first program, programNumber=1 for second and so on.
3. Get a program from **getNextProgram**(). Read its priority number and based on the priority of the program, select the heap for the program.
4. Once the heap is decided, insert the program into the heap depending on its ProgramAge. The program with the highest ProgramAge should be at the top of the heap.
5. Also, write the ProgramAge, programNumer and creationTime of this program in a text file named “**progseq.txt**”. This file has the order in which the program were CREATED.
6. Repeat steps 4-6, exactly 6 times.
7. And finally, once all the programs are in their respective heaps, start processing the programs by getting a program from the heap one at a time moving from high to low. Pass each selected program to a function **ProcessThis(program p)**, that is a dummy function and only prints the contents of the program in a file called “**config.txt**”: This file has the order in which the program were PROCESSED.

**Format of file:** file should look something like:

High:

ProgramNumber = 2, ProgramAge=3.5

ProgramNumber = 4, ProgramAge=2

…

Mid:

ProgramNumber = 0, ProgramAge=13.2

…

Low:

ProgramNumber = 3, ProgramAge=4.0

ProgramNumber = 1, ProgramAge=3.0

…

**ALERT! THIS IS AN APPLICATION OF HEAP. DO NOT MODIFY THE HEAPADT FOR THIS PROGRAM.**