Exercise 2 – Implementing a Heap

Due by the end of your Week-3 laboratory class. (2 marks)

This exercise is to be done during your week 3 laboratory class. When you complete the exercise show your work to your lab tutor to have your work marked. The marking is based mainly on correct implementation and code readability. You should implement your code in one file (e.g. ex2.cpp, ex2.c, ex2.java). Make sure your program has a header comment block containing the name of the exercise, your name and your student login (e.g. jfk01). You may implement your solution in C, C++, Java or Python.

For this exercise you are to implement a heap, using an array as shown in the lectures. The heap will be built by repeatedly:

1. Reading an input from a file (in this case an integer)
2. Adding it after the last element of the current heap
3. Sift it up into the correct position, to make a min heap.

At the end of all the inputs you are to print out the first five elements of the min heap array. You should then convert the array into a max heap by calling makeheap and again print the first 5 elements. Your heap should be able to hold 100 integers. A pseudo-code outline for the program is given below:

```
Begin main
    display a prompt for the file name
    read in the file name
    try to open the file
    if the file fails to open
          print an error message on the screen and exit
    fi
    while we can read an int from the file
          insert the int into the array (to make a min heap)
    elihw
    close the file
    print the first 5 elements of the heap
    makeheap() (to make a max heap)
    print the first 5 elements of the heap
End main
```

Do not implement the heap using a class or with STL. The heap must be implemented using a fixed size array of ints (200 entries should be enough). The heap array and the number of items in the heap should be global variables (or private static variables in Java).

Information on the above functions is available from the week 2 lecture notes.

When you are finished, test your program using the provided text file named “ex2.txt” and show your code and the output to your lab tutor to receive your marks. Also, submit your file via banshee using the submit command below.

```
$ submit –u login –c CSCI203 –a ex2 filename
```

where '*login*' is your UNIX login ID and ‘*filename*’ is the name of your file.

If you are unable to attend your lab class and demonstrate your work on time due to circumstances beyond your control (e.g. sickness), contact your lecturer to request an extension.