DATA STRUCTURES PRACTICESHEET

Name: ___________________________

Student ID :________________________________

STACKS AND QUEUES

QUESTION A: What are some applications of Stacks

QUESTION B: What is the difference between a Stack, Queue and Array

QUESTION C: Implement a stack and a queue using a single array, the array should store both items of the queue and stack

QUESTION D: Convert this expression from infix to postfix
(1+3)-(10/8)/3-9*2

QUESTION E: Convert this expression from postfix to infix
1 9 8 * + 7 5 - + 3 +

QUESTION F: Implement both expression conversion techniques
postfix => infix, infix => postfix

QUESTION: Write a function which evaluates an infix expression,
evaluate(num1, operator, num2)
evaluate(1, +, 3) *returns 4*
evaluate(4, /, 2) *returns 2*

QUESTION G: Implement the the valid parenthesis code

QUESTION E: Write a function which implements round robin scheduling of an operating system.You are given an array of apps, each app contains a name, os running time, and time left.

```
Struct app {
        string name
        int time_left
        int run_time
}
```

Your output should be a print of a apps currently running on the os until all apps have been run.

QUESTION F: Write the enqueue function for a priority queue.

QUESTION G: Write a function which removes all the even numbers from a queue.

QUESTION H: Implement the reversal of a Queue and Stack recursively

HASHING

QUESTION A: Describe and implement all the collision handling techniques

QUESTION B: Insert this numbers into an array using double hashing

10, 12, 35, 22, 50

Hash1 = x % size

Hash2 =7 + x % size

SORTING

QUESTION A: Sort these numbers using, merge, quick and shell sorting algorithms

6, 10, 25, 4, 9, 23, 11, 19, 1, 16, 19

QUESTION B: Describe the complexity of the above algorithms

QUESTION C: Write and explain the pseudocode for implementing quick and mergesort

HEAPS

QUESTION A: Insert this numbers into a max heap and show the state of the heap for each insertion

3, 9, 7, 2, 25, 1, 7, 12, 8, 14, 6