CSE 373 – Data Structures
Homework 2

Assigned: Wednesday, April 10, 2002

Due: Wednesday, April 17, 2002

At the start of class

Remember to also do an electronic turnin of stack.c.

Your name: _____

Student number: _____

1. What are the following values?

    a. $\log_2 256$

    b. $2^4 - 1$

    c. 2^{32}

    d. $\log_2 32$

2. Show the following relationships are true.

    a. $f(n) = 1 + 2 + 3 + \ldots + n$ is $O(n^2)$

    b. $f(n) = n!$ is $O(n^n)$. Note: n! is "n factorial", defined as $n(n-1)(n-2)\ldots(3)(2)(1)$.

    c. $f(n) = 5x^2 + 4x\log_2 x$ is $\Theta(x^2)$

3. As described in the project writeup, the stack function push just terminates the program if there is a stack overflow condition. This is not the most user-friendly approach to this problem. A better method is usually to trap the problem in the push function, and increase the amount of memory allocated for the stack (the buffer). Read the descriptions of the memory management routines malloc and realloc, and write another version of push that handles stack overflow by expanding the size of the stack area. Write the new function here.

4. Consider the code that you wrote for the stack functions and also the code that uses it in mainStack.c and postfix.c. One Stack S is created in mainStack.c, and then S is passed to both ConvertToPostfix and EvaluatePostfix for them to use while working. They push different types of items on the Stack, but it all works.

    a. Function ConvertToPostfix pushes pointers to NULL terminated strings onto the stack. Draw a picture similar to slide 14 from lecture 6 showing what your stack looks like after ConvertToPostfix has pushed one string. Show the correct values of capacity and current, arrows to show the various pointers, and a labeled block for the string.

b. Function EvaluatePostfix pushes pointers to small structs that each contain a single numeric value (a double) onto the stack. Draw a picture similar to slide 14 from lecture 6 showing what your stack looks like after EvaluatePostfix has pushed two operand structs. Show the correct values of capacity and current, arrows to show the various pointers, and labeled blocks for the operand struct.