Dylan Buchheim

Group N

Lab 502 / Lecture 004

Questions:

I had no questions.

Acknowledgement:

I was the only one who contributed to my code.

//stack.c Pseudocode:

//Type definitions: Node, node is a struct type which holds an integer n, and node pointer \*next.

\*Stack, is a pointer struct which holds a node pointer \*head.

//Function Definition:

Function Name: Create

Input: N.A.

Output: s, the stack created by the function.

Data: s, a stack.

Plan: //P1: allocate memory for stack s, malloc().

//P2: initialize the head of the stack to NULL.

//P3: return s.

//Function Definition:

Function Name: IsEmpty

Input: s, a stack.

Output: 1, if the stack is empty.

0, if the stack is not empty.

Data: N.A.

Plan: //P1: if s->head == NULL, return 1.

//P2: otherwise return 0.

//Function Definition:

Function Name: Push

Input: s, a stack.

n, the integer you want to push into s.

Output: N.A.

Side Effect: n is appended to the front of the stack.

Data: \*tempNode, a node pointer.

Plan: //P1: allocate memory for tempNode, malloc().

//P2: initialize tempNode->n = n.

//P3: initialize tempNode->next = s->head.

//P4: set the head of s to tempNode.

//Function Definition

Function Name: Pop

Input: s, a stack.

Output: n, the integer which was on top of the stack.

Side Effect: the top node of the stack is deleted.

Data: \*h, a node pointer.

n, the integer which is popped off the stack.

Plan: //P1: set \*h to the head of s.

//P2: set n = h->n.

//P3: move the head of s to the next node in the stack.

//P4: free the memory held by h, free().

//P5: return n.

//Main Pseudocode:

Data: s, a stack.

Plan: //P1: create the stack s, s = Create().

//P2: call GetSeries(s).

//P3: call PrintSeries(s).

//Function Definition:

Function Name: GetSeries

Input: s, a stack.

Output: N.A.

Side Effect: pushes the users input into the stack.

Data: n, the integer which stores the users input.

Plan: //P1: print the prompt for the user to enter a series of numbers.

//P2: open an infinte loop.

//P2.1: get user input for n.

//P2.2: if n is not = 0, Push(s,n).

//P2.3: otherwise break from the loop.

//Function Definition:

Function Name: PrintSeries

Input: s, a stack.

Output: N.A.

Side Effect: of the printing the integers in the stack in reverse order.

Data: N.A.

Plan: //P1: print the prompt saying that “the series you gave is…”

//P2: start an infinte loop.

//P2.1 if IsEmpty(s) is not = 1, print Pop(s).

//P2.2: otherwise break from the loop.