

COP 3502 Quiz #2 Version A (Recursion, Linked Lists, Stacks, Queues, SLMP)

Last Name: _____, First Name: _____

Circle Your Recitation: F6:30 T1:30 T2:30 T3:30 T4:30

1) (5 pts) In the Towers of Hanoi code shown in class, a transcript of the moves to solve the puzzle was printed. Write a related function that adds up the sum of the disk numbers for each move for solving the puzzle for n disks. For example, if $n = 3$, your function should return 11 because the disks that move are disk numbers 1, 2, 1, 3, 1, 2 and 1, respectively and $1 + 2 + 1 + 3 + 1 + 2 + 1 = 11$. Notice that the sum to the left of the three and the right of the three are the same, so it's okay to calculate it once and multiply that by 2.

```
int sumTowerDisks(int n) {
```

```
}
```

2) (5 pts) Evaluate the following postfix expression, showing the state of the stack at each of the indicated points.

3 5 2 2 ^A * - 3 2 ^B * + ^C * 9 + 5 /

A

B

C

Value of the Expression: _____

3) (8 pts) Consider using a linked list of digits to store an integer. Specifically, the integer 2471 would be stored as the following linked list: $1 \rightarrow 7 \rightarrow 4 \rightarrow 2$. (Thus the first node stores the units digit, the second node the ten's digit, the third node the hundred's digit and so forth.) Write a recursive function that takes in a pointer to the front of a linked list storing an integer and returns the numeric value of the list. (Hint: To solve the problem carefully think about the value generated by the list $7 \rightarrow 4 \rightarrow 2$ in relation to the number above. Also, an empty list has a value of 0.)

```
typedef struct numNode {
    int digit;
    struct node* next;
} numNode;

int getValue(numNode* number) {

}
```

4) (5 pts) What problem does the function below solve? Namely, what is the return value of the function in relation to the input parameters? Your answer should be a regular sentence or two in English and NOT a literal translation of the code. The latter will receive **no credit!**

```
// Pre-condition: array is size n and 1 <= k <= n.
int mysteryA(int* array, int n, int k) {

    int res = 0, tmp;
    for (int i=0; i<k; i++)
        res += array[i];
    tmp = res;

    for (int i=0; i<n-k; i++) {
        tmp = tmp - array[i] + array[i+k];
        if (tmp > res) res = tmp;
    }

    return res;
}
```

3, 7, 6, 2, 9, 4, 8, 5, 1, 0

_____ , _____ , _____ , _____ , _____ , _____ , _____ , _____ , _____ , _____ , _____

Assume that the `queueSize` is initially set to 10 and that the following code segment is run:

```
for (int i=0; i<9; i++) {
    enqueue(MyQueuePtr, i+1);
    enqueue(MyQueuePtr, 2*(i+1));
    int tmp = dequeue(MyQueuePtr);
}
```

front _____ numElements _____ queueSize _____

[illegible]

Scratch Page - Please clearly label any work on this page you would like graded.