CSCI 2380 Practice Final Exam

- Do not start until instructed to do so.
- Write your UTRGV ID only in the space provided at the top of each page.
- The midterm is closed - no books, notes, computers, cell phones, calculators, etc.
- The time allotted for the exam is 70 minutes.
- There are 7 questions worth 28 points total; each problem is worth 4 points.

<table>
<thead>
<tr>
<th>Problem</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1
Problem 1. What does the following program print? Draw a box around your solution.

```cpp
#include <iostream>
using namespace std;

void yas(int a)
{
    if (a == 4)
        cout << 'a' << endl;
    else
    {
        for (int i = 1; i <= a; ++i)
            yas(a+1);
    }
}

int main()
{
    yas(1);
}
```

Problem 2. Write a loop-less function named evens that takes an int named x and prints the even numbers from 2 to 2x on the same line. E.g., evens(5) prints "2 4 6 8 10".
Problem 3. For the following questions, consider only theoretical, worst-case running time.

Adding an element to a binary search tree takes $\Theta(\log(n))$ time.  □ True □ False

Merge sort takes $\Theta(n^2)$ time. □ True □ False

Insertion sort is slower than quick sort. □ True □ False

Searching in a balanced binary search tree takes $\Theta(\log(n))$ time. □ True □ False

Problem 4. Complete the recursive function that computes the number of leaves in a binary tree.

```c
int leaf_count(Node* root)
{
    if (root == 0)
        return _____ ;

    int sc = 0;

    sc += leaf_count( __________ );
    sc += leaf_count( __________ );

    if (root->left _____ 0 && root->right _____ 0)
        return _____ + sc;

    return _____ + sc;
}
```

3
Problem 5. Draw the binary search tree resulting from inserting the following strings in the given order: "dog", "bat", "noon", "farm", "zoo", "apple", "say", "hip".
Problem 6. Consider the singly linked list Queue implementation we have seen in class. Complete the following method that returns the number of elements in the queue larger than $x$ and smaller than $y$.

```cpp
int Queue :: inrange(int x, int y)
{
    Node* cur = head;

    int count = _____ ;

    while ( cur _____ 0 )
    {
        if ( cur->n _____ x _____ cur->n _____ y)
        {
            count = count _____ 1 ;
        }
        cur = _____->_____ ;
    }

    return count;
}
```
Problem 7. Consider the doubly linked list Deque implementation we have seen in class. Complete the following method for this class: a method that removes the element at the back of the deque and places it at the front of the deque.

```cpp
void Deque :: cutsies()
{
    if ( head _____ tail )
        return;

    if ( head _____ 0 )
        return;

    Node* new_head = _____ ;
    Node* new_tail = tail->_____ ;
    head->prev = _____ ;
    new_tail->next = _____ ;
    tail->next = _____ ;
    tail->prev = _____ ;
    head = new_head;
    tail = new_tail;
}
```