

Name: \_\_\_\_\_

## CS 161 Week 6 Worksheet:

### More References vs. Pointers, Recursion

#### Post-Mid Term I:

1. "Reverse" Vocabulary!

As the material in this course become more and more abstract, the concepts become harder to grasp quickly. Instead of trying to guess the difficult concepts, we want you to provide us with terms or concepts you don't think you understand yet, and provide your current definition or understanding of the term or concept to clear any misconceptions or affirm your knowledge. Your answers can be either specific vocabulary terms or more general ideas. We'll try to address some of these concepts and/or terms in future lecture/exercises.

2. What did you struggle with the most on Exam 1?

#### References vs. Pointers:

1. Explain these terms:

- Reference
- Pointer
- Dereference

2. Answer the following questions from your conceptual point of view, and justify them using code.

1) For references and pointers:

- i. How they are created?
- ii. How to access another place of memory?
- iii. Can we access the memory location of pointers/references?

## Recursion

1. Explain these terms:

- Iteration
- Recursion
- Base case
- Recursive call

2. Trace the following code. What will each piece of code do or print? (Show your work)

```
1) int a (int i) {  
    return i + a(i - 1);  
}  
cout << a(4) << endl;
```

```
2) int b (int i) {  
    if (i == 0)  
        return 1;  
    return i * b(i - 1);  
}  
cout << b(4) << endl;
```

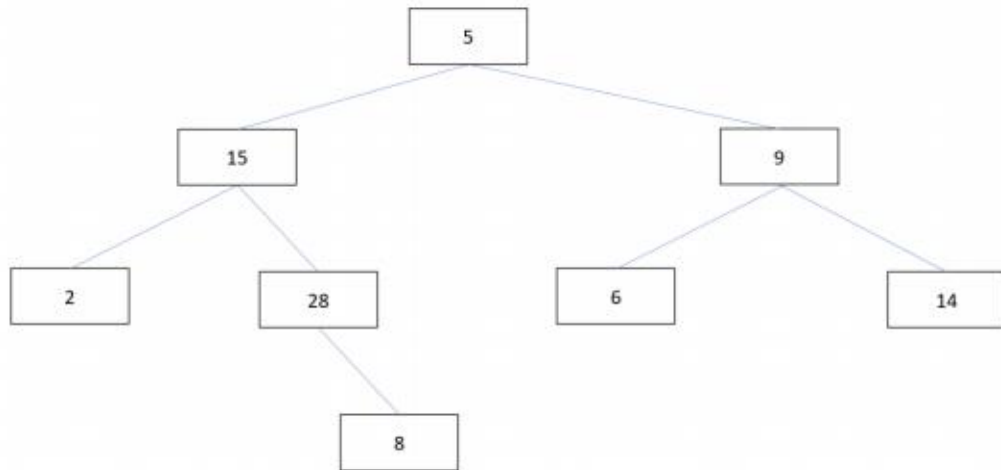
3. Write a print\_to\_zero(int i) function that will print from i to 0 Iteratively.

```
void print_to_zero(int i){  
  
  
  
}
```

4. Write a print\_to\_zero(int i) function that will print from i to 0 Recursively.

```
void print_to_zero(int i){  
  
  
  
}
```

5. See if you can figure out what the following recursive algorithm will print based off the binary tree (Don't be intimidated, try tracing through the algorithm)



```
printValue(square x)
    if square x does not exist
        return //(Don't print anything)
    printValue(x's left child)
    Print: Value of x
    printValue(x's right child)
```

6. Think of different ways you could arrange the last three lines in the recursive algorithm above and how does it change the order of the values printed.