You may use one previously prepared 8.5×11 inch piece of paper for reference material. Show all of your work clearly in the space provided or on the additional page at the end of the exam. If the additional page is used, be sure to clearly label the content for each problem. Be sure to *read each problem carefully*. You should answer all 5 questions. Note: The exam is double-sided.

1. (20 points) Using inheritance, write the definition for an adaptor class for a **stack** data structure that uses a list of `int`s for the base class. Note: **you do not need to implement the class**.

2. This is a multi-part question related to your first laboratory assignment. For this problem, suppose that the dictionary contains n words and the encrypted file contains m words. When answering these questions, be sure to list any assumptions you make and **explain the reasons behind your answers**.

(a) (10 points) If 5% of the words in the encrypted file have punctuation in them and m is much less than n, give the time complexity for your program using big-oh notation.

(b) (10 points) If all of the words in the encrypted file have punctuation in them and $m = n$, give the time complexity for your program using big-oh notation.

(c) (10 points) Suppose that n and m are both larger than 1000 and do not contain any punctuation. Suppose further that your computer can only store 100 words in memory at a time. Instead of storing all of the dictionary words in a list and all of the encrypted words in a vector, the words from both files are read into memory only when needed. Give the time complexity for your program using big-oh notation.

3. (15 points) Consider the following function.

```
#include <vector>
#include <cassert>
using std::vector;

double findKthLargest(const vector<double>& nums, unsigned int k)
{
  unsigned int N = nums.size();
  assert(N>0);
  double largest = nums[0];
  unsigned int index = 0;
  for(unsigned int i=0; i<k; ++i) {
    for(unsigned int j=1; j<N-i; ++j) {
      if(nums[j]>largest) {
        largest = nums[j];
        index = j;
      }
    }
    swap(nums[N-i], nums[index]);
  }
  return largest;
}
```

Assume that swap(double& x, double& y) is an $O(1)$ function that swaps the values found in x and y. Using big-oh notation, describe the overall worst case time complexity for the function. Be sure to explain your reasoning and state any additional assumptions that you make. Hint: Use N and k to characterize the input size.

4. Recall that in the second laboratory assignment you are asked to write a program that validates HTML and XML files. The technique you should use is to read the file and parse all of the HTML or XML tags. If the tag is a starting tag, you should place it on a stack. If the tag is an ending tag, you should verify that the top element on the stack matches the ending tag. If it does, you remove it from the stack. Otherwise, the file fails the validation test.

(a) (10 points) What should be used to characterize the input size of this program?

- Number of characters in the file
- Number of words in the file
- Number of tags in the file
- Other (please specify)

Be sure to justify your answer.

(b) (10 points) Using big-oh notation and the input size from part **(a)**, describe the overall worst case time complexity for the HTML/XML validator. Be sure to explain your reasoning.

5. (15 points) Prove, using mathematical induction, that the sum of powers of 2 is one less than the next higher power. That is, for any nonnegative integer n:

$$\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$$

Additional work area for any problem. Clearly identify to which problem the work on this page is related.