



Closed book/closed notes. 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. (10 points) Briefly explain the purpose of an adaptor class.

2. (15 points) Discuss the role of generic algorithms in the Standard Template Library. What characteristics must a function have in order to be a generic algorithm.

3. Consider the following function.

```
#include <list>
2 #include <string>

4 using namespace std;
int midterm(list<string> words)
6 {
    list<string>::const_iterator itr = words.begin();
8     int count = 0;
    while(itr!=words.end()) {
10         for(int i=0; i<(*itr).size(); ++i) {
            string word = (*itr).substr(0,i);
12             if(words.find(word)) {
                ++count;
14             }
        }
16         ++itr;
    }
18     return count;
}
```

(a) (15 points) Using big-oh notation, describe the overall worst case time complexity for the function. Assume that the words contained in the list are no more than 50 characters long. Be sure to explain your reasoning and state any additional assumptions that you make.



(b) (10 points) Using big-oh notation and the same assumptions as in part (a), describe the overall worst case time complexity for the function if the container used was a **vector** instead of the list. Be sure to explain your reasoning.



4. A palindrome is a phrase that reads the same forward and backward. Here are some examples:

- radar
- Straw? No, too stupid a fad. I put soot on warts.

Notice that punctuation, capitalization, and spacing are ignored when determining whether a phrase is a palindrome or not.

(a) (15 points) Use the `stack` and `queue` data structures to write a function that indicates whether or not the `string` that is passed to it is a palindrome. You may write a `o++` program or *clearly* describe your algorithm in English/pseudocode.



(b) (10 points) Using big-oh notation, describe the overall worst case time complexity for your algorithm. Be sure to explain your reasoning.

5.

(a) (15 points) Write a recursive function, `power`, which accepts a `double` and an `unsigned int` as input and returns the value of the `double` raised to the `unsigned int` power.

(b) (10 points) Using big-oh notation, describe the overall worst case time complexity for your algorithm. Be sure to explain your reasoning.



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



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