



Closed book/closed notes. Be sure to *read each problem carefully*. You should answer all five questions. Note: The exam is double-sided.

1. (10 points) List the three STL data structures that the `stack` can use as its underlying container.

2. (10 points) Indicate which of the three data structures (from your answer to the previous question) cannot be used as the underlying container for the `queue` class, and explain precisely why not.

3. (10 points) Recall that the definition of big-oh notation is:

An algorithm is $O(g(n))$ if there exists some constants c and n_0 , such that:

$$f(n) < c \cdot g(n), \text{ for all } n > n_0$$

where $f(n)$ is a function that describes the exact execution time of some algorithm with input size n .

Concisely and precisely explain the role of c and n_0 in the definition. I.e., What do they do? Why are they needed?

4. Consider the following class.

```
class List : public std::list<double> {
public:
    ... // constructors, destructor, assignment operator
    double& operator[](unsigned int index);
    const double& operator[](unsigned int index) const;
    // no data members
};
```

(a) (15 points) Implement the `const` version of the subscript operator for the `List` class.

Consider the following function:

```
void copy(std::vector<double>::const_iterator itr, unsigned int N, std::vector<double>& data)
{
    for(unsigned int i=0; i<N; ++i) {
        data.push_back(*itr);
        ++itr;
    }
}
```

(b) (10 points) Give the worst case asymptotic time complexity for this function **and** concisely and precisely explain your reasoning.

(c) (10 points) Suppose the `std::vector<double>` in the above function were replaced with `List`. Would the function work? If so, give the worst case asymptotic time complexity for this function **and** concisely and precisely explain your reasoning.

Consider the following function:

```
double maximum(const std::vector<double>& data)
{
    assert(data.size()!=0);
    double max = data[1];
    unsigned int sz = data.size() // assume this is a O(1) operation
    for(unsigned int i=0; i<sz; ++i) {
        if(max<data[i]) {
            max = data[i];
        }
    }
    return max;
}
```

(d) (10 points) Give the worst case asymptotic time complexity for this function **and** concisely and precisely explain your reasoning.

(e) (10 points) Suppose the `std::vector<double>` in the above function were replaced with `List`. Would the the function work? If so, give the worst case asymptotic time complexity for this function **and** concisely and precisely explain your reasoning.

5. (15 points) Suppose we have a $2 \times n$ checkerboard (two rows and n columns). Write a recursive function **domino()** that, given n , will calculate the number of possible ways in which we can cover the board with 1×2 dominos. Just to get you started:

Number(1) = 1



Number(2) = 2



Number(3) = 3

