# COP 3530
# Data Structures

## Midsemester Exam

Name: _____

## October 17, 2000

This exam has 4 questions. Each question starts on a new page. Each page is worth 50 points. Please answer each question on its page. You may write on the back of a page.

1. This question refers to the function `canSum` defined below. `canSum` returns `true` if there are two integers in the container `a` that sum to exactly `k`. Ignore all syntax errors.

```
// Container is an STL container that stores ints
template <class Container>
bool canSum( const Container & a, int k )
{
    typename Container::const_iterator itr1, itr2;

    for( itr1 = a.begin( ); itr1 != a.end( ); ++itr1 )
    {
        itr2 = itr1; ++itr2;
        for( ; ++itr2 != a.end( ); ++itr2 )
            if( *itr1 + *itr2 == k )
                return true;
    }
    return false;
}
```

   (a) Which of the following would be acceptable parameters for the `Container` type: `list<int>`, `vector<int>`, `set<int>`, `map<int,int>`? For unacceptable parameters, indicate the error.

   (b) Explain why `itr2=itr1;++itr2;` is used instead of `itr2=itr1+1;`.

   (c) What is the Big-Oh running time of this function? Does your answer depend on the actual type of `Container`?

   (d) Suppose it takes 64 seconds to run `canSum` on a `list<int>` of 100000 elements on a computer $A$. Computer $B$ is 16 times faster than computer $A$. How large a problem can be solved on computer $B$ in 8 seconds?

2. Suppose we have a map that maps names to ages:

```
map<string,int> people;
```

For space efficiency, it is decided to store the keys as basic `const char *` (C-style strings). Give the new declaration for the map, keeping in mind that since C-style strings are compared using `strcmp` (`strcmp(s1,s2)` returns a number less than 0 if `s1` is less than `s2`), you will have to instantiate the `map` with a function object, and provide the implementation of that function object.

3. You need to store 10,000 collections of **various sizes**. In each collection are `string` objects. Combined, the 10,000 collections contain 1,000,000 `string`s, whose total length is 12,000,000 characters, but some collections have very few items, while others have many. If space is the only factor, should each of the 10,000 collections be represented using a `vector<string>` or a `list<string>`? You must perform calculations that estimate the space used for both scenarios.

4. (a) Complete the implementation of the following class template. Since this is a timed exam, you **do not have to error check**. Note that the data member is a `set`. Although `set` does not have a `front` or `back` operation, you can get iterators representing `begin()` and `end()`, and you can call `insert()` and `erase()` (`erase` takes an **iterator**).

(b) Give the running time of each operation.

```
template <class Comparable>
class DoubleEnded
{
  public:
    bool isEmpty( ) const
      {                                                    }
    void makeEmpty( )
      {                                                    }
    void insert( const Comparable & x )
      {                                                    }
    const Comparable & findMin( ) const
      {                                                    }
    const Comparable & findMax( ) const
      {                                                    }
    void deleteMin( Comparable & x )
      {                                                    }
    void deleteMax( Comparable & x )
      {                                                    }

  private:
    set<Comparable> items;
    typedef set<Comparable>::iterator iter;
    typedef set<Comparable>::const_iterator citer;
};
```