Standard Data Structures

Vector

Java

Imports

```
import java.util.Vector;
```

Method Contents

```
// Initialize Vector
Vector<String> myVector = new Vector<String>();
// Push Object to the back of the list
myVector.add("String0");
myVector.add("String1");

// Traverse myVector's Contents
for (String vectorElement : myVector)
{
    System.out.println(vectorElement);
}

myVector.contains("String0");
```

C++

Header Includes

```
#include <vector>
#include <algorithm>
using namespace std;
```

Method Contents

```
// In c++ you can use primitive types within your data-structures
vector<int> myVector = new vector<int>();
myVector.push_back(0);
myVector.push_back(1);

for (vector<int>::iterator it = myVector.begin(); it != myVector.end(); ++it)
{
    cout << *it << endl;
}

// Is element present?
vector<int>::iterator it;
it = find(myVector.begin(), myVector.end(), 0);
if (it != myVector.end()) { cout << "Vector contains " << *it  << endl;}
```

Map

Java

Imports

```
import java.util.Map;
import java.util.HashMap;
import java.util.Iterator;
```

Method Contents

```
Map<String,Integer> myHashMap = new HashMap<String,Integer>();
myHashMap.put("String0", 0);
myHashMap.put("String1", 1);
myHashMap.put("String2", 2);

// Assign new value to key
myHashMap.put("String0", 3);
myHashMap.put("String1", 4);
myHashMap.put("String2", 5);

// Iterate through map's contents -- key and value
Iterator it = myHashMap.entrySet().iterator();
while (it.hasNext())
{
    May.Entry myPair = (Map.Entry) it.next();
    myPair.getKey();
    myPair.getValue();
}

// Iterate through map's contents using for-each
for (Map.Entry<String, Integer> myPair : myHashMap.entrySet())
{
    myPair.getKey();
    myPair.getValue();
}

// Get Specific Value based on key "String0"
myHashMap.get("String0");

// Check to see if map contains key
myHashMap.containsKey("String0");

// Check to see if map contains Value
myHashMap.containsValue(0);
```

C++

Header Includes

```
#include <map>
#include <string>
#include <algorithm>
using namespace std;
```

Method Contents

```
map<string, int>  myMap = new map<string, int>();
myMap["string0"] = 0;
myMap.insert(pair<string,int>("string1", 1));
for (map<string,int>::iterator it = myMap.begin(); it != myMap.end(); ++it)
{
    it->first; // Key
    it->second; //Value
}
```

Practice Problems

Problem 1

A palindrome is a word, phrase, number, or other sequence of symbols or elements, whose meaning may be interpreted the same way in either forward or reverse direction. Given a string determine whether or not it is a palindrome. Your program should print the string followed by " is palindrome" or " is not palindrome".

```
Example
Input: mom Output: mom is palindrome
Input: dog Output: dog is not palindrome
```

Problem 2

Write a program to encrypt and decrypt a message using a barrel cypher. A barrel cypher has a key with n integers. Given a key and a string you should encrypt and then decrypt the message.

```
Example Input:
123
hello world
82739
acm programming practice
Example Output:
igomq#xqumf :: hello world
iet#yzqnujuopqp(rydl|kjh :: acm programming practice
```