Name: ______________________________

USC NetID (e.g., `ttrojan`):______________________

CS 455 Final Exam
Spring 2016 [Bono]

May 11, 2016

There are 7 problems on the exam, with 76 points total available. There are 10 pages to the exam (5 pages double-sided), including this one; make sure you have all of them. There is also a one-page double-sided code handout that accompanies the exam. If you need additional space to write any answers, pages 9 and 10 of the exam are left blank for that purpose. If you use these pages for answers you just need to direct us to look there. *Do not detach any pages from this exam.*

Note: if you give multiple answers for a problem, we will only grade the first one. Avoid this issue by labeling and **circling** your final answers and crossing out any other answers you changed your mind about (though it's fine if you show your work).

Put your name and USC username (a.k.a., NetID) at the top of the exam. Also put your NetID at the top right of the front side of each page of the exam. Please read over the whole test before beginning. Good luck!

Problem 1 [8 points]

[C++] Suppose we are developing some linked list functions in C++ (called f1, f2, f3, and f4). The functions are currently defined and used in the following single-file program (some details left off so it all fits here.) The different program elements are numbered at right for your convenience. *(problem continued on the next page)*

```
#include <iostream>                   (1)


using namespace std;                  (2)


struct Node {
 int data;
 Node *next;                            (3)
 Node(int item);
 Node(int item, Node *n);
};

typedef Node * ListType;          (4)


Node::Node(int item) { . . .}
                                          (5)
Node::Node(int item, Node *n) { . . .}


// Note: the following 4 functions do no I/O
void f1(ListType list) { . . . }

void f2(ListType list) { . . . }              (6)

int f3(ListType list) { . . . }

void f4(ListType & list, int n) { . . }

int main() {
   ListType myList;
   . . .
   f1(list);
   f4(list, 3);                       (7)
   cout << f3(list);
   f2(list);

   return 0;
}
```

Problem 1 (cont.)

In the space provided below show what would be in each file for a version of this program that can use separate compilation. That is, we want to be able to compile the module with the list functions (to be in `listFuncs.h` and `listFuncs.cpp`) separately from the program that uses them. We're calling that client program `testListFuncs.cpp`. Show everything that would need to go in each file, such that, for things that were in the original file, you can **use the identifying numbers from the previous page** (i.e., you do not need to rewrite the code from the previous page) to indicate the contents that those numbers refer to. For your convenience, we provided the `#ifndef` – stuff that goes in header files. Also, we already showed where (5) would go.

```
#ifndef LIST_FUNCS_H
#define LIST_FUNCS_H









#endif
```

`listFuncs.h`

```


(5)  // we did this part for you


```

`listFuncs.cpp`

```





```

`testListFuncs.cpp`

Problem 2 [7 points]

Give the big-O worst case time to solve each of the following problems (put your answers in the space provided to the left):

_________ 1. compute the minimum and maximum values in an array of size *n*

_________ 2. search for a value in a sorted array of size *n*

_________ 3. reverse the values in an array of size *n*

_________ 4. do `list.get(i)` where `list` is a Java `LinkedList` object of size *n*

_________ 5. do `list.get(i)` where `list` is a Java `ArrayList` object of size *n*

_________ 6. use insertion sort to sort *n* values in an array

_________ 7. determine whether a linked list is already sorted

Problem 3 [4 points]

Suppose you are implementing your own Stack class. For each of the following, answer one of: front, end, or doesn't matter.

Part A. For a singly-linked list representation of a Stack, it would me most efficient to keep the top of the stack at which end of the linked list? [Note: a singly-linked list is the kind we've been implementing in C++]

Part B. For a partially-filled array representation of a Stack, it would me most efficient to keep the top of the stack at which end of the partially-filled array?

Problem 4 [2 points]

Explain one benefit of implementing and testing a subset of a class (or program) before implementing the rest of the class (or program). (2 sentences maximum)

Problem 5 [15 points]

[Java] Write a *recursive* Java method to find the number of occurrences of a particular value in an array. A solution that doesn't use recursion will receive little to no credit. Hint: you will need a helper method to do the actual recursion.

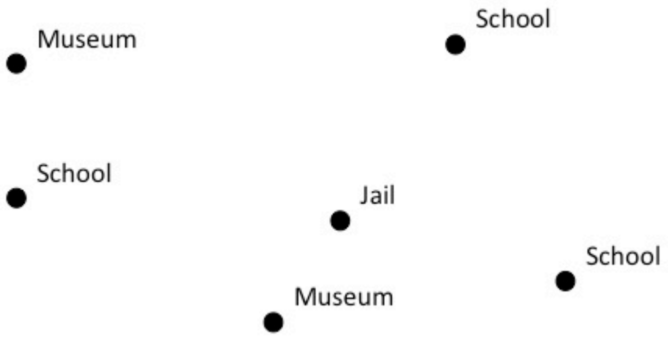
Examples (array shown as comma-separated values with brackets):

nums	***val***	***countVals(nums,val)***
[4,5,3]	5	1
[-4,3,-10,3]	3	2
[2]	3	0
[]	5	0
[5,5,5,5]	5	4

```
// returns the number of occurrences of the value val in the array nums
// (uses recursion)
public static int countVals(int[] nums, int val)
```

Problem 6 [20 points]

[Java] Suppose we want to annotate a map of a city with interesting sights. We might want to know the locations of museums or schools, for example. Here's an example of such data as it might appear on a map (other map features are not shown):



This problem concerns implementing a class to store the data for the sights. For the purposes of this problem we'll assume that any given map location can only have *one* label: e.g., location `(3207, 1245)` is a `School`. But the same label can occur at multiple locations: e.g., there may be three schools in town, thus 3 different points have the label `School`, as in the example above. Also, we'll assume that locations on a map will be represented using the Java `Point` class (that is not be a realistic way to represent a geographic location, but we'll use it here to simplify the problem), and that their coordinates can have any value, including negative values.

Our `Sights` class will have three methods in addition to the constructor: one for adding a new labeled location, one for getting the label for a location, and one for getting all the locations that have a particular label. This and the next page have the exact specification of these methods.

Implement the `Sights` class. For full credit, your representation/implementation should be as efficient as possible, taking advantage of Java library classes we have used this semester.

Note: The code handout has more information about the Java library, including the `Point` class.

```
public class Sights {
   // put private data here:









   public Sights() {         // creates an empty Sights object









   }
```

[class definition continued on the next page]

Problem 6 (cont.)

```
   // if this location isn't already part of these Sights, add the given
   // (loc, label) pair to the Sights, and returns true.
   // otherwise, returns false, and Sights object is unchanged
   public boolean add(Point loc, String label) {


  }

  // returns the label for this location or null if this location isn't in the Sights
  public String getLabel(Point loc) {


  }

  // returns the list of locations that have this label
  // or null if there are no sights with this label
  public ArrayList<Point> getPoints(String label) {


  }
}
```

Problem 7 [20 points]

[C++] Write the C++ function `compress`, which takes a linked list of `int`s passed by reference, and replaces each run in the list with a single value. A run is a sequence of two or more of the same value all next to each other. Your code should have no memory leaks (this includes reclaiming any memory no longer used).

The `Node` and `ListType` definitions are on the code handout.

Examples:

list	***list** after call to **compress(list)***
`()`	`()`
`(1)`	`(1)`
`(3 2 1)`	`(3 2 1)`
`(1 1 1)`	`(1)`
`(3 2 2 3)`	`(3 2 3)`
`(2 2 3 3 3 3 1 2 2)`	`(2 3 1 2)`

```
// PRE: list is a well-formed linked list.
void compress(ListType & list)
```

Extra space for answers or scratch work. (DO NOT detach this page.)

If you put any of your answers here, please write a note on the question page directing us to look here. Also label any such answers here with the question number and part, and circle the answer.

Extra space for answers or scratch work (cont.) (DO NOT detach this page.)

If you put any of your answers here, please write a note on the question page directing us to look here. Also label any such answers here with the question number and part, and circle the answer.