CSCE 2014 – Final Exam
Spring 2012

Student Name: _______________________

Student UAID: _______________________

Instructions: This is closed book exam. Students are allowed one 8.5x11 page of notes, but no calculators or other electronic devices.
Consider the following C++ code.

```cpp
int fun(int num)
{
    cout << num << " ";
    if (num < 0)
        return( fun(-num) );
    else if (num == 0)
        return( 0 );
    else if (num > 0)
        return( 1 + fun(num / 2) );
}

int main()
{
    int num;
    cin >> num;
    fun(num);
}
```

(4 points) What are the terminating conditions for this recursive function?

A) if (num < 0)  
B) else if (num == 0)  
C) else if (num > 0)  
D) A) and B) above.  
E) A) and C) above.

(4 points) What will the program output if the user enters ‘8’?

A) 8 8 8 8 8 …  
B) 8 4 2 1  
C) 8 6 4 2 0  
D) 8 4 2 1 0  
E) None of the above.

(4 points) What will the program output if the user enters ‘-7’?

A) -7 -7 -7 -7 -7 …  
B) -7 -3 -1 0  
C) -7 7 3 1 0  
D) 7 3 1 0  
E) None of the above.
Consider the following C++ code.

class node
{
public:
    string str;
    node* next;
};

void insert_head(node* &head, node* &tail, string str)
{
    node* temp = new node();
    temp->str = str;
    temp->next = head;
    if (head == NULL) tail = temp; // Line A
    head = temp;
}

void insert_tail(node* &head, node* &tail, string str)
{
    node* temp = new node();
    temp->str = str;
    temp->next = NULL;
    if (tail == NULL) head = temp; // Line B
    else tail->next = temp;
    tail = temp;
}

void test_head()
{
    node* head = NULL;
    node* tail = NULL;
    insert_head(head, tail, "one");
    insert_head(head, tail, "two");
    insert_head(head, tail, "three");
}

void test_tail()
{
    node* head = NULL;
    node* tail = NULL;
    insert_tail(head, tail, "one");
    insert_tail(head, tail, "two");
    insert_tail(head, tail, "three");
}
(4 points) Draw a diagram to show what the linked list will look like after the function “test_head” has been executed.

(4 points) Draw a diagram to show what the linked list will look like after the function “test_tail” has been executed.

(4 points) What is the purpose of line A) in the “insert_head” function?

A) To set the head pointer to NULL.
B) To set the tail pointer to NULL.
C) To initialize the head pointer when inserting data into an empty list.
D) To initialize the tail pointer when inserting data into an empty list.
E) None of the above.

(4 points) Write the function “print” that outputs the data in the linked list starting at the head and going to the tail. Be sure to declare all parameters and variables.
-- Stacks --

(4 points) Assume you are implementing an array-based stack of size 10 and the integer “top” contains the location of the top of the stack. How would you check for stack overflow before adding data to the stack?

A) if (top == 0) cout << “stack overflow\n”  
B) if (top < 9) cout << “stack overflow\n”  
C) if (top == 9) cout << “stack overflow\n”  
D) if (top > 9) cout << “stack overflow\n”  
E) None of the above.

(4 points) Assume you are implementing a stack using an existing linked list class with the top of the stack at the head of the list. How would you implement the stack pop operation?

A) Call the insert_head operation.  
B) Call the insert_tail operation.  
C) Call the delete_head operation.  
D) Call the delete_tail operation.  
E) None of the above.

(4 points) Assume that you are using a stack to evaluate postfix expressions, and you are reading numbers and +, -, *, and / operations from the user one at a time. What should you do when you encounter an operator?

A) Push the number onto the stack.  
B) Push the operator onto the stack.  
C) Pop two numbers off the stack and push the operator.  
D) Pop two numbers off the stack and push(num2 op num1).  
E) None of the above.

(4 points) Which of the following stack operations can cause a stack underflow?

A) Push.  
B) Pop.  
C) Top.  
D) IsEmpty.  
E) IsFull.
----- Queues -----

(4 points) What is the difference between a queue and a stack?

A) Stacks require dynamic memory, but queues do not.
B) Stacks use two ends of the structure, queues use only one.
C) Queues require dynamic memory, but stacks do not.
D) Queues use two ends of the structure, stacks use only one.
E) There is no difference between Stacks and Queues.

(4 points) Assume you are implementing an array-based circular queue, and the front index always contains the location of the front of the queue, and the end index always contains the location of the end of the queue. Which indices will change value when you do an insertion into a non-full queue?

A) Front.
B) End.
C) Both.
D) Neither.

(4 points) Assume you are implementing an array-based circular queue, and you keep track of the front index and end index. Which indices will change value when you do a deletion from a non-empty queue?

A) Front.
B) End.
C) Both.
D) Neither.

(4 points) Assume you have an integer queue class called “Queue” and you execute the following sequence of commands. Draw a diagram to show what the final queue will contain.

```java
Queue q;
q.insert(2);
q.insert(5);
q.insert(1);
q.insert( q.remove() + 1);
```
--- Sorting ---

(4 points) Given an integer array containing \{3, 1, 5, 2, 7, 3\}, what will the array contain after one pass over the array using the Bubble Sort algorithm? (Assume we are sorting the data in increasing order from left to right)

A) \{1, 2, 3, 5, 7\}.  
B) \{1, 3, 2, 5, 3, 7\}.  
C) \{1, 3, 5, 7, 3, 2\}.  
D) \{7, 5, 3, 3, 2, 1\}.  
E) None of the above.

(4 points) Which of the following sorting algorithms has an O(N log₂N) order of performance in the worst case?

A) Merge Sort.  
B) Quick Sort.  
C) Heap Sort.  
D) A) and B).  
E) A) and C).

(4 points) Which of the following sorting algorithms has an O(N^2) order of performance in the average case?

A) Insertion Sort.  
B) Bubble Sort.  
C) Selection Sort.  
D) All of the above.  
E) None of the above.

(4 points) Suppose that you are asked to sort 1,000,000 integers that have random values between 0 and 999. What sorting method would be the fastest?

A) Quick Sort.  
B) Merge Sort.  
C) Bucket Sort.  
D) Radix Sort.  
E) Bubble Sort.
--- Binary Trees ---

(4 points) Assume that you are given a complete binary tree with L levels, with the root of the tree at level one. What is the maximum number of nodes that can be stored in the tree?

A) $L$
B) $L^2$
C) $2^L$
D) $2^L - 1$
E) None of the above.

(6 points) Assume we are given the following binary search tree.

```
21
/   \
17   43
   /   \
  9    30
 /     /
2 15   41

What values would be printed if we output the tree in preorder?

What values would be printed if we output the tree in inorder?

What values would be printed if we output the tree in postorder?
(8 points) Assume that we are given the following C++ declarations as part of a binary search tree implementation. Write the C++ code necessary to complete the recursive “Search” function to find a node in the binary search tree. Assume this function is called with the root of the tree, and the tree is properly constructed.

class Node
{
    public:
        int value;
        Node * left;
        Node * right;
};

Node * Search(int item, Node * node)
{
    // item was found in tree

    // item was not found in tree

    // node value is larger than item

    // node value is smaller than item

}
---- Heaps ----

(4 points) Assume we have an array-based representation of a heap with the root at position 1. Where is the right child of the heap node at location $i$ located in array?

A) $i / 2$
B) $i + 1$
C) $2 \times i$
D) $2 \times i + 1$
E) None of the above.

(4 points) Assume we are given the following heap and wish to store this data in an array-based representation. Which of the following arrays of integers correctly represents the heap from position 1 to $N$?

![Heap Diagram]

A) $\{22, 44, 12, 81, 53, 71\}$
B) $\{81, 44, 71, 22, 12, 53\}$
C) $\{81, 44, 22, 12, 71, 53\}$
D) $\{81, 71, 44, 22, 53, 12\}$
E) None of the above.

(4 points) Starting with the original heap, draw the heap after deleting one value from the heap. Show your work.
(4 points) Starting with the original heap, draw the heap after inserting the value 77 into the heap. Show your work.

---- Hash Tables ----

(4 points) Assume that we have a hash table that uses linear probing to resolve collisions. If the table is 256 long, what is the maximum number of entries that can be placed in the hash table?

A) 256
B) 511
C) 512
D) 1024
E) There is no pre-defined maximum.

(4 points) Suppose we have implemented a hash table with linear probing to resolve collisions. The key values being stored are positive integers, and the hash function is given by “hash = key % size”. What will the hash table contain if the table size is 10, and we insert the following key values {41, 63, 150, 71, 52, 10, 4} in this order?

A) 10, 71, 52, 63, 4, 41, 150, -1, -1, -1
B) 150, 41, 71, 63, 52, 10, 4, -1, -1, -1
C) 10, 71, 52, 63, 4, -1, -1, -1, -1, -1
D) 150, 41, 52, 63, 4, -1, -1, -1, -1, -1
E) 41, 63, 150, 71, 52, 10, 4, -1, -1, -1

(4 points) What is the best definition of a collision in a hash table?

A) Two entries have different keys and different hash values.
B) Two entries with the exact same key have different hash values.
C) Two entries with different keys have the same exact hash value.
D) The hash table becomes full.
E) None of the above.

(4 points) If you are given a hash table that is implemented using linear probing, how many probes will be needed to find an empty location when we insert a new value into a hash table that is 1024 long that is 33% full?

A) 1
B) 2
C) 1.5
D) 1.33
E) 10