CIS 2168 2012 Fall Data Structures Midterm exam 10/16/2012

Name:___________________________

Problem 1 (30 points)

1. Suppose we have an array implementation of the stack class, with ten items in the stack stored at data[0] through data[9]. The CAPACITY is 42. Where does the push member function place the new entry in the array?
   A. A. data[0] B. data[1]
   B. C. data[9] D. data[10]

2. In the linked list implementation of the stack class, where does the push member function place the new entry on the linked list?
   A. At the head
   B. At the tail
   C. After all other entries that are greater than the new entry.
   D. After all other entries that are smaller than the new entry.

3. In the linked-list version of the stack class, which operations require linear time for their worst-case behavior?
   A. is_empty B. peek
   C. pop D. push
   E. None of these operations require linear time.

4. What is the value of the postfix expression 6 3 2 4 + - *:
   A. Something between -15 and -100
   B. Something between -5 and -15
   C. Something between 5 and -5
   D. Something between 5 and 15
   E. Something between 15 and 100

5. What does a call to the super() method do, and when would you use it?

6. What is the difference between private and protected visibility?

7. How are single-linked, double-linked, and circular lists different?

8. Inserting a node into a single-linked list with n nodes always takes O(n). True or False. Explain.

9. A recursive implementation of a particular method is usually easier to conceptualize than an iterative implementation of the same method, and runs faster since it generally uses fewer lines of code

10. Is linear search is an O(n^2) algorithm? Explain.

11. What kind of list is best to answer questions such as "What is the item at position n?"
    a. Lists implemented with an array.
    b. Doubly-linked lists.
    c. Singly-linked lists.
    d. Doubly-linked or singly-linked lists are equally best

12. What is the primary purpose of a constructor?
    a. To allow multiple classes to be used in a single program.
    b. To copy an actual argument to a method's parameter.
    c. To initialize each object as it is declared.
    d. To maintain a count of how many objects of a class have been created.

13. Suppose I have int b = new int[42]. What are the highest and lowest legal array indexes for b?

    a. 0 and 41 b. 0 and 42
    c. 1 and 41 d. 1 and 42

14. In a Linked list, variable nodeRef references the node A. Write a command to delete the successor node of node A. (assume successor exists)

15. What is the Big-O for the single Linked list get operation
    a. O(1)
    b. O(n)
    c. O(n^2)
    d. O(nlogn)

Problem 2 (9 questions 45 points)

1. Consider the following code

```
 Stack<Character>  s = new Stack<Character>();
String word = "carpets";
int i = 0;
while (i < word.length())
{
    s.push(word.charAt(i));
    i++;
}
while(!s.empty())
{
    System.out.print(s.peek());
    s.pop();
}
```

What is written to the screen?

A. serc B. carpets

C. steprac D. ccaarrppeettss

2. Consider the following code

```
Queue<Character>  s = new LinkedList<Character>();
String word = "carpet";
int i = 0;
while (i < word.length())
{
    s.offer(word.charAt(i));
    i++;
}
while(!s.isEmpty())
{
    System.out.print(s.peek());
    s.poll();
    if(!s.isEmpty()) s.poll();
}
```

What is written to the screen?

A. cre B. carpets

C. steprac D. ccaarrppeettss

3. Here is an INCORRECT pseudo code for the algorithm which is supposed to determine whether a sequence of parentheses is balanced:

```
declare a character stack
while ( more input is available)
{
     read a character
    if ( the character is a '(' )
            push it on the stack
  else if ( the character is a ')' and the stack is not empty )
            pop a character off the stack
  else
           print "unbalanced" and exit
```

```
   }
   print "balanced"
```

Which of these unbalanced sequences does the above code think is balanced?

A. ((())   B. ())(()
C. (()()))   D. (()))()

4. I am going to execute this code with THREE pushes and ONE pop:

```
stack<int> s = new stack<int>();
s.push(1);
s.push(2);
s.push(3);
s.pop( );
```

Suppose that s is represented by a linked list. Draw the state of the linked list after the above code:

head-----→

5. What is the time complexity (Big-O) of the following method?

```
   public static int search(int[] x, int target)
   {
      for(int i = 0; i < x.length; i++)
      {
           if(x[i] == target)
                return I;
      }

      return -1;
}
```

6. If you have a linked list head--- "A"->"B"->"C"->"D" and wrote the following two lines of code, what would the effect be?

```
Node<String> tempNode = head.next;
head.next = tempNode.next;
```

7. Assuming that substring(int index) returns a reference to a string containing all characters in the string starting at *index, is* the following method a valid recursive implementation?

```
public static int length(String str)
{

      return 1 + length(str.substring(1));

}
```

8. Draw a picture of memory after these statements:
```
 int[ ] m = new int[2];
 m[0] = 0;
m[1] = 1;
int[ ] p = m;
p[0] = 4;
p[0] = 5;
```

9. Use a stack to compute the CORRECT value of the following postfix expression. Each value is pushed on the stack, and each operation is performed on the top stack. Show the stack at times indicated below. 10 45 34 +52-* =___________

Stack before add operation

top		bottom

Stack before subtract operation

top		bottom

Stack before multiply operation

top		bottom

Stack after multiply operation

top		bottom

Problem:
Write a recursive method to test if a give string is palindrome or not.
Boolean palindrome(int s, int t, String str)
{

}
Problem:
We have two integer linked lists. A and B, each has n nodes.
A: head---1→3→15→7→…→9→null;
B: head---2→14→6→39→…→11→null;
Write a method that returns the head to the linked list C. Each node in C is the sum of the corresponding node in A and B. The length of A,B, and C are same.
C: head---3→17→21→46→…→20→null;

Problem:
Write a method “bool contains(int x, Node head)”. It returns true if the linked list referenced by “head” contains the integer x. Otherwise returns false;

```
Bool contains (int x, Node head)
{

}
```

Problem:

Give the sequence of integers printed by a call to foo(6):

```
  public static void foo(int n) {
    if (n <= 0) return;
    print(n);
    foo(n-2);
    foo(n-3);
    print(n);
  }
```