Multiple choice questions: [5 points each]

1. Suppose we have the int variables x=0, y=0, and z=2. What are the values of x, y, and z after executing the statement
   \[ x = y = z++; \]
   a) x==3, y==3, and z==3  // 2 points for this close guess
   b) x==2, y==2, and z==2
   c) x==2, y==2, and z==3
   d) x==0, y==2, and z==3
   e) x==0, y==0, and z==3

2. Java uses which parameter passing technique?
   a) by force
   b) by reference  // this another technique, for which 2 points
   c) by and large
   d) by instantiation  // no...
   e) by value

3. To initialize a String array names to store the three Strings “Huey”, “Duey” and “Louie”, you would do
   a) String names = {“Huey”, “Duey”, “Louie”};
   b) String[ ] names = {“Huey”, “Duey”, “Louie”};
   c) String[ ] names = new String{“Huey”, “Duey”, “Louie”};

4. Given that the character ‘c’ was stored in an Object by the statement
   \[ \text{Object obj = new Character(’c’);} \]
   what is the correct way to recover this value in Java 1.4.2?
   a) char letter = obj.parseChar();
   b) char letter = (char) obj;
   c) char letter = (Character) obj;
   d) char letter = ((Character) obj).charValue();
   e) char letter = (char) ((Character) obj);
5. [15 points] Suppose that we want to print out all the digits of a number called val, with each digit on a separate line. For example, if val=210, then 2 is on the first line, 1 on the second, and 0 on the third. A nice way to do this recursively, is outlined as follows:

- if val<0, do nothing.
- if val<10, print val (using “System.out.println”).
- otherwise recursively call the procedure to print the digits of val/10, and then print val%10 (using “System.out.println”).

```java
public static void printDigRec(int val){
    if (val<0) return;
    if (val<10) System.out.println(val);
    else {
        printDigRec(val/10);
        System.out.println(val%10);
    }
}
```
6. [15 points] You are given a two-dimensional array A of type integer. Its declaration and instantiation is as follows:

```java
int[][] A = new int[num_rows][num_cols];
```

Calculate and return a double which is the average of all values stored in the array.

```java
public static double twoDimAverage(int[][] A) {
    int sum = 0;
    int count = 0;
    for (int i=0; i<num_rows; i++)
        for (int j=0; j<num_cols; j++) {
            sum += A[i][j];
            count++;
        }
    if (count>0)
        return ((double) sum)/count;
    return 0;  // catchall, in case count==0
}
```
7. [15 points] On one of the assignments, you had to convert a string to an array of char. The problem here is to take a string $s$ and put each of its characters into an ArrayList. The first character of $s$ should go into the initial position, and the last character of $s$ should go into the end of the ArrayList. Recall that you cannot put primitive types directly into such a list – you must put each char into its wrapper class.

```java
public static ArrayList splitIntoArrayList(String s) {

    ArrayList split = new ArrayList();

    for (int i=0; i<s.length(); i++)
        split.add(new Character(s.charAt(i)));

    return split;

} //splitIntoArrayList
```

Note: check the last page for some useful methods.
8. [15 points] You are given an array called scores, which is instantiated and filled with test scores: score[i] contains an int between 0 and 99, representing the score of the student i on a test. We want to count the number of scores in certain ranges (in order to build a histogram). So, you will define an array
   
   int[] hist = new int[10];

to contain these counts. The goal is that hist[i] contain the number of test scores in the range 10i to 10i+9. That is
   
   • hist[0] will be the number of scores between 0 and 9,
   • hist[1] will be the number of scores between 10 and 19,
   • ...
   • and hist[9] will be the number of scores between 90 and 99.

   public static int[] buildHist(int[] scores) {
       int[] hist = new int[10];

       for (int i=0; i<scores.length; i++)
           hist[ scores[i]/10 ]++;

       return hist;
   }
9. [15 points] Write a method which will take an input String s, and return a String which is s reversed and with all occurrences of ‘A’ and ‘a’ removed.

For example, if s=“Ala Moana Mall” then return “llM noM l”.

```java
public static String badAreverse(String s) {
    String rev = “”;
    for (int i=s.length()-1; i>=0; i--)
        if ((s.charAt(i)!=’a’) && (s.charAt(i)!=’A’))
            rev += s.charAt(i);
    return rev;
}
```
10. [20 points] We maintain a *singly* linked list with a head pointer (head points to the first node). We are *not* using the LinkedList class, but are building the structure from scratch. Complete the three undone methods below:

```java
public class LinkedListExample
{
    public class Node{
        int info;
        Node next;

        public Node(int info, Node next) {
            this.info = info;
            this.next = next;
        }
    }

    public Node head;

    public LinkedListExample()
    {head = null;}

    public void addToFront(int val)
    {
        head = new Node(val, head);
    }

    public void addAfter(Node p, Node q)
    {
        // place node q after node p, even if q is null

        if (p==null) return;

        if (q! = null)
            q.next = p.next;

        p.next = q;
    }
}
```
public void removeFromFront()
{
    if (head != null)
        head = head.next;
}

public void removeAfter(Node p)
{
    //remove the node that comes after node p

    if (p==null) return;

    if (p.next != null)
        p.next = p.next.next;
}

public int size(int bound)
{
    // count the number of nodes in the list containing a value
    // which is <= bound.

    int count = 0;
    Node current = head;

    while (current != null) {
        if (current.info<bound) count++;
        current = current.next;
    }

    return count;
}
Some useful methods for question #7. You don’t need them all.

From ArrayList:

void add (int index, Object element)
   Inserts the specified element at the specified position in this list.

public boolean add( Object o)
Appends the specified element to the end of this list.

Object get(int index)
   Returns the element at the specified position in this list.

int indexOf(Object elem)
Searches for the first occurrence of the given argument, testing for equality using
the equals method.

boolean isEmpty()
Tests if this list has no elements.

Object remove(int index)
   Removes the element at the specified position in this list.

From Character:

public Character(char value)
Constructs a newly allocated Character object that represents the specified char value.