Practice Test #4 – CMPS 161

1. For each of the following situations, write a Java declaration of a single variable (array or scalar -- you decide) to handle it. Semicolons are optional, and no comments are required.

a.) The number of hairs on the head of a member of a class of junior-high-school students. (There may be up to 30 members in a class.)

b.) The number of pages in any book in a library of up to 1,000,000 books.

c.) The price (in dollars and cents) of each car on a sales lot. The sales lot has 275 parking spaces.

d.) The letter grade (on a simple A, B, C, D, F scale) of a student.

e.) Up to 50 street addresses stored as strings.

f.) The middle initials (no periods) of all the employees in a company which employs no more than 200 people at any time.

g.) The number of electoral votes received by the elected president each term from 1952 thru 1992, inclusive.

2. For each of the following situations, write a Java declaration of an initialized array to handle it. Semicolons are optional, and no comments are required.

a.) All the characters in the alphabet that are vowels.

b.) The names of the days of the week.

c.) The number of days in each month of a leap year.

3. Create the expressions described below using correct array variable references. Don’t worry about whether the subscript should take the 0’th element into account or not -- just choose an interpretation and use it consistently. You may assume that the following declarations exist:

```java
int[] computer_bought = new int[1000]; // The stock number of the computer bought by each of our customers
int[] num_comps = new int[20]; // The number of each type of computer in our stock
float[] val_comps = new float[20]; // The value of one unit of each type of computer in our stock
```

a.) The stock number of the computer bought by customer #450

b.) The name of computer type 6

c.) The total value of all the type 2 computers in our stock

d.) The number of computers in our stock that are the same as the computer bought by customer #75
3. What would be printed to the screen by the following program?

```java
void main()
{
    char [] letters = new char[33];
    String univ = "SOUTHEASTERN LOUISIANA UNIVERSITY";
    int count;
    for (count = 0; count < letters.length; count++)
        letters[count] = univ.charAt(count);
    for (count = 0; count < letters.length / 2; count++)
        swap(letters, count, letters.length - count - 1);
    print("AFTER MANIPULATION: ");
    for (count = 0; count < letters.length; count++)
        print(letters[count]);
    println();
}

void swap(char [] allLetters, int letter1, int letter2)
{
    char temp;
    temp = allLetters[letter1];
    allLetters[letter1] = allLetters[letter2];
    allLetters[letter2] = temp;
}
```

5. A linear search of parallel arrays might be set up as follows:

```java
void main()
{
    final int MAX_EMPS = 100;                   // Maximum number of employees
    int [] empId = new int[MAX_EMPS];           // ID number for each employee
    float [] empHourly = new float[MAX_EMPS];   // Hourly rate for each employee
    int numEmps,                                // Actual number of employees
        targetId,                               // somewhere in here, the parallel arrays empId and empHourly get
        currentEmp = 0;                          // filled, and num_emps attains the correct value
        int found = false;
        // ...
        targetId =
            readInt("What is the ID of the employee whose hourly rate is needed? ");

a.) Write the while statement that would be used in this linear search. Be sure to include the body of the while, as well as the header. Hint: It is only a few lines long.

b.) Suppose that you have found the employee id you are looking for at emp_id[8]. Show the reference needed to get that employee's hourly pay rate.

c.) What are the parallel arrays in this problem? Why are they considered parallel?
6. Given the following main method:

```java
void main()
{
    final int lineLength = 40;
    char [] wordToHiss = new char[lineLength + 1];
    char [] hissedWord = new char[lineLength * 2 + 1];
    int curChar = -1;
    println("Enter a word to hiss, one character at a time, stopping with a space: ");
    do
    {
        curChar++;
        wordToHiss[curChar] = readLine().charAt(0);
    }
    while (wordToHiss[curChar] != ' ');
    hiss(wordToHiss, hissedWord);
    print("A talking snake would probably say ");
    curChar = 0;
    while (hissedWord[curChar] != ' ')
    {
        print(hissedWord[curChar]);
        curChar++;
    }
}
```

Write the function `hiss` so that it returns the string passed to it, but with all 's' characters doubled. For example, if passed the string "seashore" it would return "sseashore". You may assume that all letters will be in lowercase.