

CMSC 131 Quiz 4 Worksheet

The next quiz for the course will be on Wed, Apr 3. The following list provides additional information about the quiz.

- The quiz will be a written quiz (no computer).
- The quiz will be in lab session.
- Closed book, closed notes quiz.
- Answers must be neat and legible.
- Quiz instructions can be found at <http://www.cs.umd.edu/~nelson/classes/utilities/examRules.html>.
- Make sure you know your section number and your TA's name.
- **You must take your quiz in your assigned lab/discussion session and not show up to a random discussion session. We will not grade quizzes taken in the incorrect session.**

The following exercises cover the material to be included in this quiz. Solutions to these exercises will not be provided, but you are welcome to discuss your solutions with the TAs or instructor during office hours. **It is recommended that you try these exercises on paper first (without using the computer).**

Exercises

1. How many objects are associated with the following declaration?
`int[] a;`
2. Can you create an array with 0 elements?
3. Write a static method that determines whether two arrays have the same elements, in the same positions. The prototype of the method is:

```
public static boolean same(int[] a, int[] b)
```
4. Write a static method that determines the sum of elements in the array that are in the range defined by **startIndex** and **endIndex**. The prototype of the method is:

```
public static int sum(int[] a, int startIndex, int endIndex)
```
5. Write a static method that returns an array with the elements in the range defined by **startIndex** and **endIndex**. The prototype of the method is:

```
public static int[] inRange(int[] a, int startIndex, int endIndex)
```
6. Write a static method that returns an array with the elements of the array parameter in reverse order. The prototype of the method is:

```
public static int[] reverse(int[] a)
```
7. Write a static method that places in the output parameter **answer**, elements in array **a** that have a value less than or equal to **upperLimit**. The prototype of the method is:

```
public static int find(int[] a, int upperLimit, int[] answer)
```
8. Write a Java program that:
 - a. Reads integer values and places them into an array named **src**.
 - b. Reads integer values and places them into array named **find**.
 - c. Prints "Yes" if **find** is a subset of **src**, that is, all elements in **find** appear in **src** (duplicates are fine).
9. Implement a program that reads values from the user, places them into an array, and initializes the second half of the array with values that are double of each of the values provided in the first half. For example, if the user provides the values 10, 40, 25, your program will create an array with the values 10, 40, 25, 20, 80, 50. Use the message "Enter number of elements:" to read the number of values provided by the user (e.g., 3 in the previous example) and "Enter values:" to read each of the values provided by the user (e.g., 10, 40, 25, in the previous example). Notice you do not need to print the array; you just need to initialize it with the expected values.

10. Write a static method called **findEvens** that places in the output parameter **answer**, elements in array **src** that are even. The method will return the number of even values that were placed in the answer array. If **src** and/or **answer** are null, no computation will take place and the method will return -1. You can assume the answer array object is large enough to fit the number of values found to be even.

```
public static int findEvens(int[] src, int[] answer)
```