Bill Manaris : Spring 2016 / CSCI 230 Homework 1

Assigned Date: Tuesday, Jan. 12, 2016
Due Date: Thursday, Jan. 21, 2016
Due Time: 5 mins before beginning of class

Last modified on January 21, 2016, at 07:12 AM (see updates)

Purpose

This assignment focuses on:

- deriving recursive solutions
- implementing recursion in Java
- Java programming style and documentation

This is a **solo** assignment. **You may discuss the assignment only with the TA or the instructor.**

Assignment

Create a Java program, `Recursion.java`, which contains the following recursive methods:

1. `boolean isMember(int[] list, int item, int startIndex)`, which returns `true` if `item` is in `list` starting at `startIndex`, `false` otherwise.
2. `int numberItems(int[] list, int startIndex)`, which returns the number of items (inclusive) starting from `startIndex` all the way to the end.
3. `int numberItemsReverse(int[] list, int startIndex)`, which returns the number of items (inclusive) starting from `startIndex` all the way to the beginning.
4. `int countItem(int[] list, int item, int startIndex)`, which returns a count of how many times `item` appears in `list` (inclusive) starting from `startIndex`.
5. `int[] reverseList(int[] list)`, which returns a new array containing the `list` items in reverse order. The original `list` should remain unaltered.

Note

You must use the above method specifications exactly. If you deviate from them, our test code will not work, and you may receive a zero for those methods.

Input

The program should accept several lines of input, as follows:

an `int` (e.g., 1), which specifies which method to run (e.g., the first method above)
the number of integers in the list (0 or more), followed by those integers (these should be read into an array by your code)
any other parameters required by the method (one per line)

You may assume that the input is formatted properly and is valid.

Output

The program should output the value returned by the method called. Nothing more.

I/O Sample 1

For example, given this input:

```
1
5 2 3 1 0 -15
1
0
```

the output of your program should be:

```
true
```

I/O Sample 2

For example, given this input:

```
1
5 2 3 1 0 -15
1
2
```

the output of your program should be:

```
true
```

I/O Sample 3

For example, given this input:

```
1
5 2 3 1 0 -15
1
4
```

the output of your program should be:

```
false
```

and so on.

Documentation

Follow the Golden Rule of Style: "A program should be as easy for a human being to read and understand as it is for a computer to execute." [1]

In general, you should comment any variable, obscure statement, block of code, method, and class you create.

Your comments should express **why** something is being done, as opposed to **how** – the how is shown by the code.

Also see the class's Java coding standards, for additional information, including top-level documentation.

NOTE: You may use Javadoc formatting, if you prefer. If so, do it consistently.

Submissions

Submit via OAKS as follows:

1. Create a folder named 'First.Last', where 'First' is your first name, and 'Last' is your last name.
2. Place your program inside that folder.
3. Compress the folder as a .zip file.
4. Upload on OAKS.

Grading

Your grade will be based on how well you followed the above instructions, and the depth/quality of your work.

Reference

1. Cooper, D. and Clancy, M. (1985) "Oh! Pascal", 2nd ed., W.W. Norton & Company, New York, p. 42.

(Printable View of http://www.cs.cofc.edu/~manaris/?n=Spring2016.CSCI230Homework1)