Completed by:

Kestutis IT Dev - June, 2011.

Questions and Answers

1. Given two integer arrays, array A and array B, of random length, remove elements that appear in B from A. Both A and B may contain duplicates. Can you do the same with string arrays? If you are multilingual, please provide an implementation in 2 programming languages. Please comment on your implementation, and the tradeoffs you have chosen. Please provide an ability to test.

   ANSWER:

   I wrote two minimal examples in Php and Python of this task implementation.

   Implementation in Php (using LAMP server and DreamWeaver CS5.5 IDE):
   File: "*activity_n1_simple/index.php*" – works with both integer and string arrays.

   Implementation in Python (using PyCharm 1.5 IDE):
   File: "***activity_n1_simple/index.py***"

   Implementation in C#:
   Assignment required to implement this task only in 2 programming languages, but I'm also able to do that using C# and Visual Studio 2010 IDE.

   Also I wrote a bigger project on Php 5.3.X for this task.
   (~1 KLOC – about a thousand lines of code and comments)
   This project can be found here:
   File folder: "*activity_n1_extended/*"

   The algorithm can be found at:
   DublicateRemover::removeBothExistingElements

   Program entry point:
   File: *"activity_n1_extended / index.php"*

   Example unit test:

File: ***"activity_n1_extended / UnitTest.DuplicateRemover.php"***

This project includes the following functionality:

- Object-oriented design and directory structure
- Class composition and inheritance (MiniDebugger, ArrayDataValidator, DuplicateRemover)
- Derived method overriding (ModernException::__toString())
- Different visibility for methods (protected, private, public)
- Array data validator (class ArrayDataValidator)
- Algorithm debugger (I wrote a minimal MiniDebugger class, which can be enabled / disabled in index.php @ // Settings)
- Exception handler, including usage of Nested Exceptions (as of Php 5.3.0+)
- UnitTest – usually I use a PhpUnit to write unit tests and check code coverage of that unit test, but this time I didn't used it – I wrote a minimal unit test framework (class UnitTest) for writing basic class methods tests in program. Included test is not deep and only for DuplicateRemover class.
  When I'm writing a unit test, I'm mostly follow the testing structure from Visual Studio 2010 automatic unit tests generator.
  The next steps for that application would be:

- Use MVC: use "Smarty" template engine to implement template.index.php in a Smarty template file – templates/index.tpl, extend program by using model-view design patterns (i.e. "Observer")
- Use "PhpUnit" unit testing framework, and write unit tests to cover all the possible paths in a program and reach all possible exceptions.
- Use "PhpDoc" (phpDocumentor) to generate a code documentation files for use in the future.
- Include backwards-compatible code for Php 5.2.X and Php 5.1.X

2. Implement a function that reverses the contents of a character string efficiently. Please provide an ability to test.

ANSWER:

File: "*activity_n2/index.php*"

The difference becomes visible only if we apply a reverse function only for a really long string. Between various methods to concatenate strings in Php, the best results are then, when I use the "**.=**" operator.

For an extremely large data I would prefer to use a "HipHop for Php" method - transform my code into C++ code, then build a 'dictionary' of characters used for the string and build a

dynamic list, the switch “next” and “back” pointers.

3. Given the query 'SELECT * FROM t1 join t2 on t1.f = t2.f WHERE ...' -- what would be the different steps to consider in order to speed up the query?

ANSWER:

1. SELECT * - wrong solution, we will probably use only 1-4 fields of that table, not all, so for faster query we should use field names (*SELECT aaa, bbb, ccc*) or numbers (*SELECT 1, 2, 3*) instead of selecting all fields.

2. WHERE … - we should add indexes on every WHERE field. If that field has unique value, we should use a “UNIQUE” keyword. The fastest way is to use a field with a “PRIMARY KEY” in WHERE statement.

3. We should know how we want to join results. INNER JOIN, LEFT JOIN or RIGHT JOIN might be a better than FULL OUTER JOIN.

4. FOREIGN KEY – also might be used to speed up the query, if DB engine does have support for foreign keys.

4. Explain what the following code is doing; correct any problems and explain.

ANSWER:

```
// This function is for collecting all values of specific database table field (field name in MySQL DB is ‘value’)
function get_values($reqtxt) {
        // We should consider to use super globals instead of register globals,
        // which is deprecated as of Php 5.3.0
        $mysqli_link  = $GLOBALS[‘mysqli_link’]; // mysql connection
        // Get a pointer to the first row of mysql result
        $req = $mysqli_link->query($reqtxt);
        $i = 0;
        // I always want to be sure that I’m starting from an empty string
        $value_list = “”;
        // fetch, not getch. Iterate throught every mysql row and put all current mysql row fields names
        // and values as array into $rec variable.
        while ($rec = $req->fetch_assoc())
        {
                // Take a value of field ‘value’ and append it to the end of string with a list of all previous values
                $value_list .= "'".$rec['value']."'";
                // Code must be readable and follow code standards,
                // so putting everything in one line is a bad solution
                if ($i > 0)
                {
                        // Insert a separator. We should follow coding standards,
                        // and use double quotes instead of single quotes – the same as we did before
                        $value_list .= “,”;
                }
                $i++;
        }
        return $value_list;
}
```

5. Below is some basic HTML markup. Create JavaScript code to do the following: <…>

ANSWER:

File: "*activity_n5/ js / scripts.js*"

Program entry point:

File: *"activity_n5 / index.html"*

I wrote a script without using a javascript library. I prefer to use jQuery, when it is possible.