**COM 411 ASSIGNMENT**

1. Write a PHP program that prompt the user to input three numbers .The program should then output the numbers in ascending order.

<?php

// Prompt user to input three numbers

echo "Enter three numbers: \n";

$num1 = readline("Number 1: ");

$num2 = readline("Number 2: ");

$num3 = readline("Number 3: ");

// Convert input to integers

$num1 = intval($num1);

$num2 = intval($num2);

$num3 = intval($num3);

// Put numbers in an array

$numbers = array($num1, $num2, $num3);

// Sort the array in ascending order

sort($numbers);

// Output the numbers in ascending order

echo "Numbers in ascending order: " . implode(", ", $numbers) . "\n";

?>

2. Write a PHP Function ,smallestindex ,that takes as parameters an int array and its size,and returns the index of the smallest element in the array .Also,write a programt to test your function .

Here's a PHP function named `smallestIndex` that takes an integer array and its size as parameters, and returns the index of the smallest element in the array:

<?php

function smallestIndex($arr, $size) {

if ($size == 0) {

return -1; // Return -1 if array is empty

}

$minIndex = 0;

for ($i = 1; $i < $size; $i++) {

if ($arr[$i] < $arr[$minIndex]) {

$minIndex = $i;

}

}

return $minIndex;

}

?>

```

And here's a program to test the `smallestIndex` function:

```php

<?php

include 'smallestindex.php'; // Include the file containing the function

// Test the function

$array = [5, 3, 9, 2, 7];

$size = count($array);

$index = smallestIndex($array, $size);

echo "Array: " . implode(", ", $array) . "\n";

echo "Index of the smallest element: " . $index . "\n";

echo "Smallest element: " . $array[$index] . "\n";

?>

3.Write a PHP program that prompts the user to input a string and outputs the string in uppercase(Use a character array to store the string)

<?php

// Prompt user to input a string

echo "Enter a string: ";

$inputString = readline();

// Convert the string to uppercase using a character array

$charArray = str\_split($inputString);

$upperArray = array\_map('strtoupper', $charArray);

$upperString = implode("", $upperArray);

// Output the string in uppercase

echo "Uppercase string: " . $upperString . "\n";

?>

4.Write a PHP program to compute the addition of of N by M matrices.Allow the user to determine the size of the row and column

<?php

// Function to create a matrix of size N by M

function createMatrix($rows, $cols) {

$matrix = array();

for ($i = 0; $i < $rows; $i++) {

$matrix[$i] = array();

for ($j = 0; $j < $cols; $j++) {

// Prompt user to input each element of the matrix

echo "Enter element at position ($i, $j): ";

$matrix[$i][$j] = readline();

}

}

return $matrix;

}

// Function to add two matrices

function addMatrices($matrix1, $matrix2) {

$result = array();

$rows = count($matrix1);

$cols = count($matrix1[0]);

for ($i = 0; $i < $rows; $i++) {

for ($j = 0; $j < $cols; $j++) {

// Add corresponding elements of the two matrices

$result[$i][$j] = $matrix1[$i][$j] + $matrix2[$i][$j];

}

}

return $result;

}

// Prompt user to input the size of rows and columns

echo "Enter the number of rows: ";

$rows = readline();

echo "Enter the number of columns: ";

$cols = readline();

// Create the first matrix

echo "Enter elements for the first matrix:\n";

$matrix1 = createMatrix($rows, $cols);

// Create the second matrix

echo "Enter elements for the second matrix:\n";

$matrix2 = createMatrix($rows, $cols);

// Add the matrices

$resultMatrix = addMatrices($matrix1, $matrix2);

// Output the result

echo "Resultant matrix after addition:\n";

foreach ($resultMatrix as $row) {

echo implode(" ", $row) . "\n";

}

?>

5. Write a PHP program that declares an array alpha of 50 components of the type float .Initialize the array so that the first 25 components are equal to the square of the index  variable and the last 25 components are equal to three times the index variable.Output the array so that 10 elements per line are printed.

<?php

// Declare the array alpha of 50 components

$alpha = array();

// Initialize the array according to the given conditions

for ($i = 0; $i < 50; $i++) {

if ($i < 25) {

$alpha[$i] = $i \* $i; // First 25 components are equal to the square of the index variable

} else {

$alpha[$i] = 3 \* $i; // Last 25 components are equal to three times the index variable

}

}

// Output the array with 10 elements per line

$count = 0;

foreach ($alpha as $element) {

echo $element . " ";

$count++;

if ($count % 10 == 0) {

echo "\n"; // Start a new line after every 10 elements

}

}

?>