Profesor: *Neiner Maximiliano*

Parte 1 - Ejercicios Simples

Aplicación Nº 1 (Sumar números)
Confeccionar un programa que sume todos los números enteros desde 1 mientras la suma no supere a 1000. Mostrar los números sumados y al finalizar el proceso indicar cuantos números se sumaron.
Aplicación Nº 2 (Mostrar fecha y estación)
Obtenga la fecha actual del servidor (función *date*) y luego imprímala dentro de la página con distintos formatos (seleccione los formatos que más le guste). Además indicar que estación del año es. Utilizar una estructura selectiva múltiple.
Aplicación Nº 3 (Obtener el valor del medio)
Dadas tres variables numéricas de tipo entero ***$a***, ***$b*** y ***$c*** realizar una aplicación que muestre el contenido de aquella variable que contenga el valor que se encuentre en el medio de las tres variables. De no existir dicho valor, mostrar un mensaje que indique lo sucedido.
Ejemplo 1: $a = 6; $b = 9; $c = 8; => se muestra 8.
Ejemplo 2: $a = 5; $b = 1; $c = 5; => se muestra un mensaje "No hay valor del medio"
Aplicación Nº 4 (Calculadora)
Escribir un programa que use la variable ***$operador*** que pueda almacenar los símbolos matemáticos: '**+**', '**-**', '**/**' y '*****'; y definir dos variables enteras ***$op1*** y ***$op2***. De acuerdo al símbolo que tenga la variable $operador, deberá realizarse la operación indicada y mostrarse el resultado por pantalla.
Aplicación Nº 5 (Números en letras)
Realizar un programa que en base al valor numérico de una variable ***$num***, pueda mostrarse por pantalla, el nombre del número que tenga dentro escrito con palabras, para los números entre el 20 y el 60.
Por ejemplo, si $num = 43 debe mostrarse por pantalla "cuarenta y tres".

Parte 2 - Ejercicios con Arrays

Aplicación Nº 6 (Carga aleatoria)
Definir un Array de 5 elementos enteros y asignar a cada uno de ellos un número (utilizar la función *rand*). Mediante una estructura condicional, determinar si el promedio de los números son mayores, menores o iguales que 6. Mostrar un mensaje por pantalla informando el resultado.
Aplicación Nº 7 (Mostrar impares)
Generar una aplicación que permita cargar los primeros 10 números impares en un Array. Luego imprimir (utilizando la estructura ***for***) cada uno en una línea distinta (recordar que el salto de línea en HTML es la etiqueta ***
). Repetir la impresión de los números utilizando las estructuras ***while y ***foreach***.
Aplicación Nº 8 (Carga aleatoria)
Imprima los valores del vector asociativo siguiente usando la estructura de control ***foreach***:
$v[1]=90; $v[30]=7; $v['e']=99; $v['hola']= 'mundo';

Aplicación Nº 9 (Arrays asociativos)
Realizar las líneas de código necesarias para generar un Array asociativo $lapicera, que contenga como elementos: `***color***´, `***marca***´, `***trazo***´ y `***precio***´. Crear, cargar y mostrar tres lapiceras.

Aplicación Nº 10 (Arrays de Arrays)
Realizar las líneas de código necesarias para generar un Array asociativo y otro indexado que contengan como elementos tres Arrays del punto anterior cada uno. Crear, cargar y mostrar los Arrays de Arrays.

Parte 3 - Ejercicios con Funciones

Aplicación Nº 11 (Potencias de números)
Mostrar por pantalla las primeras 4 potencias de los números del uno 1 al 4 (hacer una función que las calcule invocando la función *pow*).

Aplicación Nº 12 (Invertir palabra)
Realizar el desarrollo de una función que reciba un Array de caracteres y que invierta el orden de las letras del Array.
Ejemplo: Se recibe la palabra "HOLA" y luego queda "ALOH".

Aplicación Nº 13 (Invertir palabra)
Crear una función que reciba como parámetro un string (***$palabra***) y un entero (***$max***). La función validará que la cantidad de caracteres que tiene *$palabra* no supere a *$max* y además deberá determinar si ese valor se encuentra dentro del siguiente listado de palabras válidas: "Recuperatorio", "Parcial" y "Programacion". Los valores de retorno serán:
1 si la palabra pertenece a algún elemento del listado.
0 en caso contrario.

Aplicación Nº 14 (Par e impar)
Crear una función llamada **esPar** que reciba un valor entero como parámetro y devuelva *TRUE* si este número es par ó *FALSE* si es impar.
Reutilizando el código anterior, crear la función **esImpar**.