Prueba de Evaluación Continuada 2

|  |  |
| --- | --- |
| **Módulo** | 3: Implantació d’aplicacions web en entorn internet, intranet i extranet. |

# Enunciado

Programa las siguientes funciones en PHP y escribe los test necesarios para comprobar su correcto funcionamiento. Implementa todas las pruebas utilizando el framework PHPUnit.

## Problema 1

Programar una función que busque el máximo de un array de enteros recibido como argumento.

**Ejemplo**: max([3,4,6,1,65,-3,6,10]): 65

|  |  |  |
| --- | --- | --- |
| **ENTRADA** | **SALIDA ESPERADA** | **DESCRIPCIÓN** |
| 3,4,-6,1,65,-3,6,-10,1024 | 104 | Se ha comprobado el mayor numero del array, sin hacer discriminación entre numeros positivos y negativos |

|  |  |  |
| --- | --- | --- |
| **ENTRADA** | **SALIDA ESPERADA** | **DESCRIPCIÓN** |
| 3,4,-6,1,65,-3,6.6,-10,1024,1024.1 | 1024.1 | Se ha comprobado el mayor numero del array, siendo todos positivos con decimales. |

|  |  |  |
| --- | --- | --- |
| **ENTRADA** | **SALIDA ESPERADA** | **DESCRIPCIÓN** |
| -2,-55,-43,-4 | -2 | Se ha comprobado el mayor numero del array, siendo todos negativos. |

|  |  |  |
| --- | --- | --- |
| **ENTRADA** | **SALIDA ESPERADA** | **DESCRIPCIÓN** |
| -333,-55,-43,-404,0 | 0 | Se ha comprobado el mayor numero del array, siendo todos negativos y existiendo 0. |

## Problema 2

Programar una función que busque el mínimo de un array de enteros recibido como argumento.

**Ejemplo**: min([3,4,6,1,65,-3,6,10]): -3

|  |  |  |
| --- | --- | --- |
| **ENTRADA** | **SALIDA ESPERADA** | **DESCRIPCIÓN** |
| 3,4,-6,1,65,-7,6,-10 | -10 | Se ha comprobado el menor numero del array, entre numeros positivos y negativos. |

|  |  |  |
| --- | --- | --- |
| **ENTRADA** | **SALIDA ESPERADA** | **DESCRIPCIÓN** |
| -3,-55,-43,-4 | -55 | Se ha comprobado el menor numero del array, entre numeros negativos. |

|  |  |  |
| --- | --- | --- |
| **ENTRADA** | **SALIDA ESPERADA** | **DESCRIPCIÓN** |
| -3.2,-55.2,-43.4,-55.3 | -55.3 | Se ha comprobado el menor numero del array, entre numeros negativos con decimales. |

|  |  |  |
| --- | --- | --- |
| **ENTRADA** | **SALIDA ESPERADA** | **DESCRIPCIÓN** |
| 12,25,83,73,125 | 12 | Se ha comprobado el menor numero del array, entre numeros negativos. |

|  |  |  |
| --- | --- | --- |
| **ENTRADA** | **SALIDA ESPERADA** | **DESCRIPCIÓN** |
| 12,25,83,0,73,125 | 0 | Se ha comprobado el menor numero del array, entre numeros positivos existiendo. |

## Problema 3

Dada una cadena, implementar una función que diga (true o false) si la secuencia "<->" aparece en medio. Se considerará que está en medio si el número de caracteres que hay antes y después difieren como mucho en una unidad.

**Ejemplo**: flechaMedio("xaj<->bb"): true

Nota Problema 3: Ya que no se especifica en el enunciado no se ha tenido en cuenta, en la función que pudiera pasarse en la cadena dos veces la secuencia o la repetición de alguno de elementos de dicha secuencia dentro de la cadena.

|  |  |  |
| --- | --- | --- |
| **ENTRADA** | **SALIDA ESPERADA** | **DESCRIPCIÓN** |
| cms<->dkd | true | Se comprueba que existan los elementos de la cadena y que su posición se corresponda con el medio. |

|  |  |  |
| --- | --- | --- |
| **ENTRADA** | **SALIDA ESPERADA** | **DESCRIPCIÓN** |
| cmsd<->dkt | True | Se comprueba que existan los elementos de la cadena y que su posición se corresponda con el medio difiere un valor a la derecha. |

|  |  |  |
| --- | --- | --- |
| **ENTRADA** | **SALIDA ESPERADA** | **DESCRIPCIÓN** |
| cms<->zdkt | True | Se comprueba que existan los elementos de la cadena y que su posición se corresponda con el medio difiere un valor a la izquierda. |

|  |  |  |
| --- | --- | --- |
| **ENTRADA** | **SALIDA ESPERADA** | **DESCRIPCIÓN** |
| cms<>dkd | False | Se comprueba que existan los elementos de la cadena. |

|  |  |  |
| --- | --- | --- |
| **ENTRADA** | **SALIDA ESPERADA** | **DESCRIPCIÓN** |
| cms->dkd | False | Se comprueba que existan los elementos de la cadena. |

|  |  |  |
| --- | --- | --- |
| **ENTRADA** | **SALIDA ESPERADA** | **DESCRIPCIÓN** |
| cmsdkd | False | Se comprueba que existan los elementos de la cadena. |

|  |  |  |
| --- | --- | --- |
| **ENTRADA** | **SALIDA ESPERADA** | **DESCRIPCIÓN** |
| cms-dkd | Flase | Se comprueba que existan los elementos de la cadena. |

## Problema 4

Una sección de "espejo" en una matriz es un grupo de elementos contiguos tales que en algún lugar de la matriz, el mismo grupo aparece en el orden inverso. Devolver el tamaño de la sección espejo más grande que se encuentra en la matriz dada.

**Ejemplo**: espejoMayor([1, 2, 3, 8, 9, 3, 2, 1, 5]): 3

**Ejemplo2**: espejoMayor([1, 2, 1, 4]): 3

|  |  |  |
| --- | --- | --- |
| **ENTRADA** | **SALIDA ESPERADA** | **DESCRIPCIÓN** |
|  |  |  |

# Entrega

Un archivo comprimido con el nombre ec2-m3-*nombre-alumno*.zip donde *nombre-alumno* es el nombre del alumno.

El archivo tendrá que contener este documento con las tablas rellenadas y una carpeta con todos los archivos del proyecto excepto la carpeta vendor y el archivo vendor.lock.

# Evaluación

|  |  |  |
| --- | --- | --- |
| **Problema** | **Programación** | **Pruebas** |
| 1 | 0.5 | 1.5 |
| 2 | 0.5 | 1.5 |
| 3 | 1.25 | 1.5 |
| 4 | 1.75 | 1.5 |

# Fecha límite de entrega

La fecha límite de entrega es el 03/10/2017.