

PHP developer

Nota : El tiempo de resolución de la prueba es de **2 días** como máximo después de su recepción. Una vez terminada la solución, subir el código fuente a un repositorio público de git (github por ejemplo) y brindar la URL como respuesta.

PARTE 01 : Resolver los siguientes casos usando íntegramente PHP

Consideraciones preliminares:

Problema 01

Usando PHP, crear una clase llamada **ChangeString** que tenga un método llamado **build** el cual tome un parámetro string que debe ser modificado por el siguiente algoritmo . Reemplazar cada letra de la cadena con la letra siguiente en el alfabeto. Por ejemplo reemplazar **a** por **b** ó **c** por **d**. Finalmente devolver la cadena.

Indicaciones

- Crear la solución en un solo archivo llamado **ChangeString.php**
- El método build devuelve la salida del algoritmo
- Considerar el siguiente abecedario : a, b, c, d, e, f, g, h, i, j, k, l, m, n, ñ, o, p, q, r, s, t, u, v, w, x, y, z.

Ejemplos

- entrada : "123 abcd*3" salida : "123 **bcde***3"
- entrada : "***Casa 52" salida : "*****Dbtb** 52"
- entrada : "***Casa 52Z" salida : "*****Dbtb** 52**A**"
- a

Problema 02

Usando PHP, crear una clase llamada **CompleteRange** que tenga un método llamado **build** el cual tome un parámetro de colección de números enteros positivos (1,2,3, ...n). El algoritmo debe completar si faltan números en la colección en el rango dado. Finalmente devolver la colección completa.

Indicaciones

- Crear la solución en un solo archivo llamado **CompleteRange.php**
- El método build devuelve la salida del algoritmo
- Considerar el parámetro de colecciones con números enteros positivos ordenados de manera ascendente. Ejemplo [4, 6, 7 ,10]

Ejemplos

- entrada : [1, 2, 4, 5] salida : [1, 2, 3, 4, 5]
- entrada : [2, 4, 9] salida : [2, 3, 4, 5, 6, 7, 8, 9]
- entrada : [55, 58, 60] salida : [55, 56, 57, 58, 59, 60]

Problema 03

Usando PHP, crear una clase llamada **ClearPar** que tenga un método llamado **build** que reciba como parámetro una cadena formada sólo por paréntesis `((()())(())())((()))`. El algoritmo debe eliminar todos los paréntesis que no tienen pareja. Finalmente devolver la nueva cadena.

Indicaciones

- Crear la solución en un solo archivo llamado **ClearPar .php**
- El método **build** devuelve la salida del algoritmo
- Considerar solamente cadenas formadas de paréntesis

Ejemplos

- entrada : `"()())"` salida : `"()())"`
- entrada : `"()()"` salida : `"()()"`
- entrada : `"("` salida : `""`
- entrada : `"((("` salida : `"("`

PARTE 02 : Desarrollar la siguiente aplicación

Consideraciones preliminares:

- La aplicación debe ser realizada usando íntegramente PHP
- Se debe usar el framework **slim** (<http://www.slimframework.com/>) para el desarrollo de la aplicación
- Basarse su desarrollo íntegramente en PHP
- El archivo para trabajar con el aplicativo esta en <https://goo.gl/wmV5K1>

Problema práctico

Se desea realizar un pequeño prototipo de aplicación web que tiene como finalidad registrar y evaluar a los empleados de la empresa Developers SAC. Para ello se tiene un archivo llamado **employees.json** donde se encuentra una estructura inicial de empleados. En primer lugar, se desea generar como pantalla inicial el listado de todos los empleados considerando los siguientes campos:

- name
- email
- position
- salary

Seguidamente, en el mismo listado se debe mostrar un barra de búsqueda por email.

En segundo lugar, se desea acceder al detalle de cada empleado donde se debe realizar un resumen de los siguientes campos:

- name
- email
- phone
- address
- position
- salary
- skills

Finalmente, se desea liberar un servicio web en formato XML que permita buscar los empleados de acuerdo a un rango de salario mínimo y máximo. Ejemplo : obtener todos los empleados que tienen salario entre 1000 y 1500 dólares.