DEVPLACE

Dev Challenge

BOOTCAMP JAVA

Objetivo del documento

Presentar una serie de problemas básicos que permitan evaluar los conocimientos técnicos, y la capacidad resolutiva.

Aclaraciones

Tratar de ser lo más claro posible en el código. Comenta lo más que puedas.
Por favor, si aún no lo tienes, crear un usuario de github, crear un repositorio PÚBLICO y depositar ahí el código para que podamos evaluarlo.

Si no pudiste terminar todos los ejercicios, no te preocupes, envíanos igual el link al repositorio con los ejercicios que hayas podido resolver. No es condición necesaria terminar TODO el challenge para quedar seleccionado.

Challenge Algoritmos

Lenguaje a utilizar: Javascript, Java, C, C#, Node, Phyton.

1. Pide un número por teclado e indica si es un número primo o no. Un número primo es aquel que solo puede dividirse entre 1 y sí mismo. Por ejemplo: 25 no es primo, ya que 25 es divisible entre 5, sin embargo, 17 si es primo.

2. Escribe una aplicación que solicite al usuario que ingrese una contraseña cualquiera. Después se le pedirá que ingrese nuevamente la contraseña, con 3 intentos. Cuando acierte ya no pedirá más la contraseña y mostrará un mensaje diciendo “Felicitaciones, recordás tu contraseña”. Si falla luego de 3 intentos se mostrará el mensaje “Tenes que ejercitar la memoria”. Los mensajes quedarán en pantalla a la espera que el usuario presione una tecla, luego de esto se cerrará el programa.

3. Por teclado se ingresa el valor hora de un empleado. Posteriormente se ingresa el nombre del empleado, la antigüedad y la cantidad de horas trabajadas en el mes. Se pide calcular el importe a cobrar teniendo en cuenta que al total que resulta de multiplicar el valor hora por la cantidad de horas trabajadas. Además, si el empleado tiene más de 10 años de antigüedad hay que sumarle la cantidad de años trabajados multiplicados por $30. Imprimir en pantalla el nombre, la antigüedad y el total a cobrar.

4. Generar un número aleatorio comprendido entre 0 y 1000. Pedir, a continuación, al usuario adivinar el número escogido por el ordenador. Para ello, debe introducir un número comprendido entre 0 y 1000. Se compara el número introducido con el calculado por el ordenador y se muestra en la consola "es mayor" o "es menor" en función del caso. Se repite la operación hasta que el usuario encuentra el número.

5. Pedir al usuario que ingrese un número repetidamente hasta que ingrese un -1 y en ese caso se terminará el programa.
   Al terminar, mostrará lo siguiente:
   a. – mayor número introducido
   b. – menor número introducido
   c. – suma de todos los números
   d. – suma de los números

Challenge POO

Lenguaje a utilizar: Javascript, Java, C, C#, Node, Phyton.

Haz una clase llamada **Password** que siga las siguientes condiciones:

- Que tenga los atributos **longitud** y **contraseña** . Por defecto, la longitud será de 8 caracteres.
- Los constructores serán los siguiente:
  - Un constructor por defecto.
  - Un constructor con la longitud que nosotros le pasemos (Generará una contraseña aleatoria con esa longitud).
- Los métodos que implementa serán:
  - **esFuerte()**: devuelve un booleano si es fuerte o no, para que sea fuerte debe tener más de 2 mayúsculas, más de 1 minúscula y más de 3 números.
  - **generarPassword()**: genera la contraseña del objeto con la longitud que tenga.
  - Métodos get para los atributos contraseña y longitud.
  - Método set para stributo longitud.

Crear un proyecto de consola que solicite al usuario una contraseña por teclado y muestre un mensaje indicando la contraseña y si es o no fuerte.

Challenge - Base de Datos

Lenguaje a utilizar: SQL,Javascript, Java, C, C#, Node, Phyton.

1. Generar un script sql que cree el modelo de tablas que respetar la siguiente problemática:

   Se desea conocer el nombre, apellido, teléfono y email de los empleados municipales del país, que tengan promedio un salario mayor a $70.000, y tengan una antigüedad entre 10 y 15 años. También es necesario conocer el nombre del puesto y la municipalidad a la que pertenece.
2. Llevar este modelo de base de datos a un modelo de clases.
3. Realizar una aplicación de consola que llene las tablas con datos ficticios
4. Realizar una función en dicha aplicación, que muestre la información de los empleados y municipalidades que cumplan con los requisitos explicados anteriormente.