



Objetivo del Documento

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

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.

Desde el momento en que se envía el challenge usted tendrá 72 horas para enviar el challenge y que sea aceptado por DEV PLACE.

Envío del Challenge

Ordenar el repositorio de Github del challenge de tal forma que hayan 3 directorios con el nombre de cada módulo y dentro de esos directorios habrán directorios con los respectivos nombres o números de los ejercicios con sus respectivos archivos.

Enviar mail a organizacion@devplace.tech con el asunto "Challenge Full Stack JS NOMBRE APELLIDO" en el mail tendria que figurar el link al repositorio.

Modulo Algoritmos

Puedes utilizar cualquier Lenguaje de Programación. Será valorada y destacada que el challenge esté resuelto con el lenguaje Javascript.

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 sí 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

Estructura de Datos

Listas, Pilas, Colas.

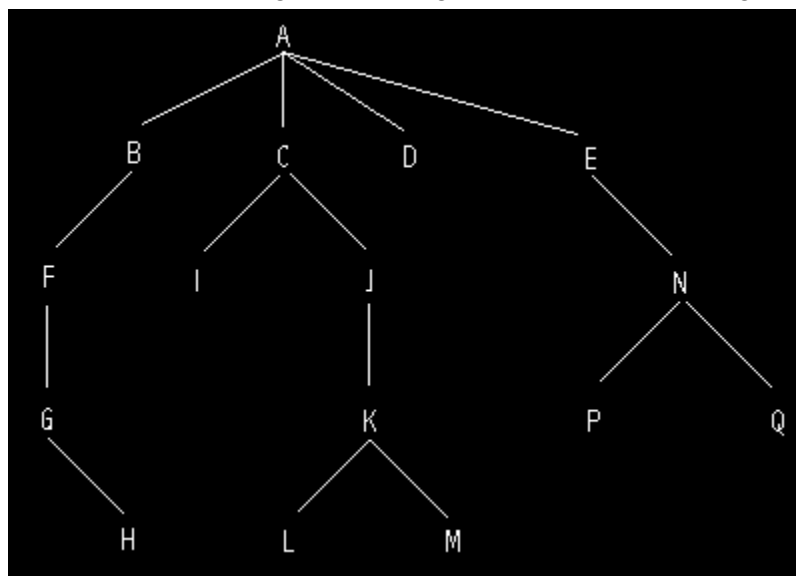
- 1) Implementar una función que recibe una lista de enteros L y un número entero n de forma que modifique la lista mediante el borrado de todos los elementos de la lista que tengan este valor

2)Escribir una función Reemplazar que tenga como argumentos una pila con tipo de elemento int y dos valores int: nuevo y viejo de forma que si el segundo valor aparece en algún lugar de la pila,sea reemplazado por el segundo.

3)Construir una función que sume los elementos de una lista de enteros recursivamente.

Arboles. General.

1)Responder a las siguientes preguntas sobre el árbol siguiente:



1. ¿Qué nodo es la raíz?
2. ¿Cuántos caminos diferentes de longitud tres hay?
3. ¿Es un camino la sucesión de nodos HGFBACI?
4. ¿Qué nodos son los ancestros de K?
5. ¿Qué nodos son los ancestros propios de N?
6. ¿Qué nodos son los descendientes propios de M?
7. ¿Qué nodos son las hojas?
8. ¿Cuál es la altura del nodo C?
9. ¿Cuál es la altura del árbol?
10. ¿Cuál es la profundidad del nodo C?
11. ¿Cuál es el hermano a la derecha de D?
12. ¿Es I hermano a la derecha de F?
13. ¿Está F a la izquierda de J?
14. ¿Está L a la derecha de J?
15. ¿Qué nodos están a la izquierda y a la derecha de J?
16. ¿Cuántos hijos tiene A?
17. Listar los nodos del árbol en preorden,postorden e inorden.

Modulo POO

Puedes utilizar cualquier Lenguaje de Programación. Será valorada y destacada que el challenge este resuelto con el lenguaje Javascript.

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 siguientes:
 - 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 atributo 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.