Examen 1er corte (40)

Escriba códigos C# para resolver lo indicado:

1. **Contador de accesos global**
   * Declara una variable global que almacene el número de veces que se ha ejecutado un método. Cada vez que se llame al método, debe incrementar el contador y mostrarlo en pantalla.
2. **Área de un rectángulo con variables locales**
   * Crea un método que pida la base y altura de un rectángulo, use variables locales para calcular el área y la imprima en pantalla.
3. **Conversor de temperatura con variable global**
   * Declara una variable global que indique la escala de conversión (ejemplo: Celsius a Fahrenheit). Haz un método que use esa variable para realizar la conversión.
4. **Suma acumulada con variable global**
   * Declara una variable global sumaTotal. Cada vez que el usuario ingrese un número (en un método), se debe sumar al acumulado y mostrar el total actualizado.
5. **Edad de una persona con variables locales**
   * Diseña un método que use variables locales para leer el año de nacimiento de una persona y calcule su edad. La edad no debe almacenarse de forma global.
6. **Juego de intentos con variable global**
   * Crea un programa que use una variable global para almacenar el número de intentos de adivinar un número secreto. Cada intento se registra y se muestra.
7. **Promedio de notas con variables locales**
   * Haz un método que calcule el promedio de tres notas introducidas por el usuario usando solo variables locales.
8. **Control de sesión con variable global**
   * Declara una variable global booleana sesionIniciada. Simula un inicio de sesión que la cambie a true cuando el usuario ingrese la contraseña correcta.
9. **Perímetro de un círculo con variables locales**
   * Escribe un método que reciba el radio como parámetro, calcule el perímetro y lo retorne usando únicamente variables locales.
10. **Juego de puntuación con variables globales**
    * Declara una variable global puntuacion. Cada vez que se llame al método GanarPuntos, la puntuación debe aumentar en 10 y mostrarse en pantalla.

**Ejercicios de evaluación propuestos**

**11. Sistema de Inventario simplificado**

* + Declara una variable global que almacene el total de productos en inventario.
  + Crea métodos locales para:
    - Agregar productos.
    - Retirar productos.
    - Consultar el inventario actual.
  + Cada acción debe actualizar la variable global y mostrar los resultados.

**12. Simulación de Cajero Automático**

* + Usa una variable global que represente el saldo de una cuenta bancaria.
  + Implementa métodos con variables locales para:
    1. Depositar dinero.
    2. Retirar dinero (validando que no exceda el saldo).
    3. Consultar saldo.

1. **Juego de Adivinanza Mejorado**
   * Usa una variable global para almacenar el número secreto.
   * Un método con variables locales debe pedir al usuario que adivine.
   * Otro método debe aumentar un contador global de intentos y mostrar un mensaje de acierto o error.
2. **Gestión de Notas de Estudiantes**
   * Declara una lista global que almacene todas las notas.
   * Métodos locales deben:
     + Agregar una nota.
     + Calcular el promedio (solo con variables locales).
     + Mostrar la nota más alta y la más baja.
3. **Calculadora Científica Básica**
   * Usa una variable global para guardar el último resultado calculado.
   * Crea métodos con variables locales para realizar:
     1. Suma, resta, multiplicación y división.
     2. Potencias y raíces cuadradas.
   1. Cada vez que se realice una operación, el último resultado debe actualizarse en la variable global.
4. **Sistema de Registro de Usuarios**
   * Crea una lista global que almacene nombres de usuarios registrados.
   * Métodos con variables locales deben permitir:
     1. Registrar un nuevo usuario.
     2. Validar si un usuario ya existe.
     3. Mostrar todos los usuarios.
5. **Simulador de Tienda en Línea**
   * Declara una variable global carritoTotal.
   * Crea métodos con variables locales para:
     + Agregar un producto (sumando su precio al total).
     + Eliminar un producto.
     + Consultar el total actual de la compra.
6. **Validador de contraseñas seguras**

Diseña un programa que valide si una contraseña ingresada cumple los siguientes requisitos:

* Mínimo 8 caracteres
* Al menos una mayúscula
* Al menos una minúscula
* Al menos un número
* Al menos un carácter especial (!@#$%^&\*)

**Requisitos:**

* Crea una función EsContraseñaValida(string pass) que devuelva true o false.
* Crea procedimientos adicionales que verifiquen cada criterio individualmente.
* Usa modularidad adecuada y variables locales/globales donde se justifique.

1. **Validación de fecha (día, mes, año)**

Desarrolla un programa que:

Pida al usuario ingresar una fecha (día, mes, año).

Valide si la fecha es válida (mes entre 1-12, días válidos según el mes, etc.).

* Determine si el año es bisiesto.
* Use funciones para cada validación (por ejemplo, EsBisiesto, DiaValido, MesValido).
* Muestra si la fecha ingresada es correcta o no.

Requisitos técnicos:

* Funciones para validaciones lógicas.
* Uso de variables locales y globales.

1. **Número capicúa (palíndromo numérico)**

Crea un programa que:

* Pida al usuario un número entero positivo.
* Determine si ese número es capicúa (por ejemplo, 121, 1331).
* Usa una función para invertir el número (no con arreglos).
* Compara el número original con el invertido.
* Usa una variable global para el número original y funciones auxiliares para la inversión.

Requisitos técnicos:

* Función con retorno.
* Sin arreglos ni manipulación de cadenas (usa operadores matemáticos).
* Uso de variables globales/locales.