

Instrucciones:

Realizar en papel el programa requerido por el enunciado. Luego pasarlo en máquina para comprobar su correcto funcionamiento y enviarlo por mensajería de Ideas a los docentes del aula. El archivo a **entregar** debe tener extensión **lpr** o **pas** y debe tener el apellido y nombre del estudiante (**perez.Juan.pas**). Tienen tiempo hasta el Jueves **08/05/2025**.

La solución debe mostrarse en la consulta de práctica del **viernes 09/05/2025**.

Ejercicio:

En la próxima evaluación global de robots humanoides, se analizarán modelos que realizan tareas humanas en diferentes categorías: manipulación de objetos, reconocimiento de voz, interacción social, etc. Cada robot será evaluado con un puntaje del 0 al 10 (entero). Al momento de la inscripción, se leerán los datos de cada robot:

- Código de robot (18 caracteres).
- ID del robot (entero).
- Nombre del robot.
- Nombre del Fabricante.
- CUIT del Fabricante (entero).
- Puntaje obtenido en cada una de las competencias anteriores en las que participó. El puntaje está en una escala de 0 a 10.

Se dispone de un arreglo de fabricantes habilitados para participar, ordenada alfabéticamente por nombre. Cada fabricante tiene asociado un atributo que indica cuántos años lleva participando en competencias globales.

Implementar:

1. Validación de ID del robot

Implementar un módulo que valide el ID del robot. Para ello desarrolle un módulo cuya tarea sea descomponer un número que recibe como parámetro y retorne la suma de sus dígitos. Un ID del robot es válido si la suma de sus dígitos es mayor que la suma de los dígitos del CUIT del fabricante.

2. Validación del código de robot

Implemente un módulo que valide el código del robot. Para ello, desarrolle una serie de submódulos que procesen los caracteres del código y analicen si se cumple con el siguiente patrón ABC:

- a) Módulo 1 (Parte A): Secuencia de 6 caracteres que deben ser letras mayúsculas y números, con al menos 2 de cada uno (ej. si tiene 5 mayúsculas y 1 número no se cumpliría la Parte A de la secuencia).
- b) Módulo 2 (Parte B): Secuencia de 5 dígitos ordenados en secuencia de menor a mayor que no estén en la Parte A.
- c) Módulo 3 (Parte C): Secuencia de 7 dígitos o letras que contengan el carácter 'A' seguido de 'B'. En el caso de tener dígitos, estos no deben estar en la Parte A.

Nota: Dejar de procesar caracteres cuando el patrón deje de cumplir.

3. Verificación de fabricante habilitado

Implemente un módulo que reciba el arreglo de fabricantes, el nombre del fabricante del robot y un valor mínimo de años de antigüedad requerido. Determine si el fabricante está en el arreglo y si cumple con el mínimo de años de antigüedad. Utilice búsqueda dicotómica.

4. Simulación de inscripción

Simule la inscripción de los robots a la competencia. Utilice el módulo implementado en 1) para validar el ID del robot, el módulo implementado en 2) para validar el código del robot y el módulo de 3) para verificar si el fabricante está habilitado y tiene más de 3 años de antigüedad. En caso de no cumplir, el robot no podrá participar de la competencia. Los robots se cargan hasta que llega el robot con nombre "DEEPLearn", el cual debe procesarse.

Implemente dos versiones (con su respectivo módulo y estructura de datos) del atributo "puntaje" teniendo en cuenta que:

- a. Se almacenan a lo sumo el puntaje de 10 competencias.
- b. Indefinidas (no se conoce de antemano la cantidad de competencias en las que participó).

5. Clasificación por puntaje

Separe la estructura creada en el inciso 3.a) o 3.b), en 11 listas de acuerdo con el puntaje (valor entero) de las competencias registradas.

Estas estructuras deben quedar ordenadas por ID del robot de menor a mayor, y almacenar el puntaje.

6. Lista doblemente enlazada

Cree una lista doblemente enlazada que combine las listas creadas en el inciso 5) e imprima en pantalla el modelo del robot y el fabricante (entre paréntesis) primero en orden ascendente y luego en orden descendente.

Observaciones adicionales:

- ✓ Asegúrese de que el código sea claro y modular.
- ✓ Modularice adecuadamente y libere la memoria de las estructuras dinámicas.
- ✓ Utilice comentarios para explicar el funcionamiento de cada módulo.
- ✓ Pruebe el programa con casos límite para garantizar su correcto funcionamiento.
- ✓ No utilizar herramientas de inteligencia artificial para la generación de código.

```
const
    MAX_FABRICANTES = 30;

type
    TFabricante = record
        nombre: string[70];
        antigüedad: integer;
    end;

    TFabricantes = array[1..MAX_FABRICANTES] of TFabricante;

const
    FABRICANTES: TFabricantes = (
        (nombre: 'AI Creators'; antigüedad: 9),
        (nombre: 'AI Dynamics'; antigüedad: 12),
        (nombre: 'AI Innovators'; antigüedad: 4),
        (nombre: 'AI Pioneers'; antigüedad: 9),
        (nombre: 'AutoGenius'; antigüedad: 11),
        (nombre: 'AutoMech'; antigüedad: 14),
        (nombre: 'CyberAndroids'; antigüedad: 7),
        (nombre: 'CyberWorks'; antigüedad: 10),
        (nombre: 'EvoBots'; antigüedad: 7),
        (nombre: 'EvoMech'; antigüedad: 8),
        (nombre: 'FutureBots'; antigüedad: 10),
        (nombre: 'FutureMinds'; antigüedad: 5),
        (nombre: 'MechSolutions'; antigüedad: 12),
        (nombre: 'NanoBots Co.'; antigüedad: 6),
        (nombre: 'NanoMinds'; antigüedad: 7),
        (nombre: 'NeuraBotics'; antigüedad: 8),
        (nombre: 'NextGen Androids'; antigüedad: 6),
        (nombre: 'NextGen Robotics'; antigüedad: 3),
        (nombre: 'Quantum Robotics'; antigüedad: 5),
        (nombre: 'QuantumAndroids'; antigüedad: 4),
        (nombre: 'RoboGenius'; antigüedad: 13),
        (nombre: 'RoboInnovators'; antigüedad: 3),
        (nombre: 'RoboMasters'; antigüedad: 6),
        (nombre: 'RoboTech Inc.'; antigüedad: 15),
        (nombre: 'SmartAndroids'; antigüedad: 8),
        (nombre: 'SmartBots'; antigüedad: 5),
        (nombre: 'SynthMind'; antigüedad: 9),
        (nombre: 'SynthTech'; antigüedad: 8),
```

```
(nombre: 'TechHumanoid'; antigüedad: 11),  
(nombre: 'TechPioneers'; antigüedad: 10)  
);
```