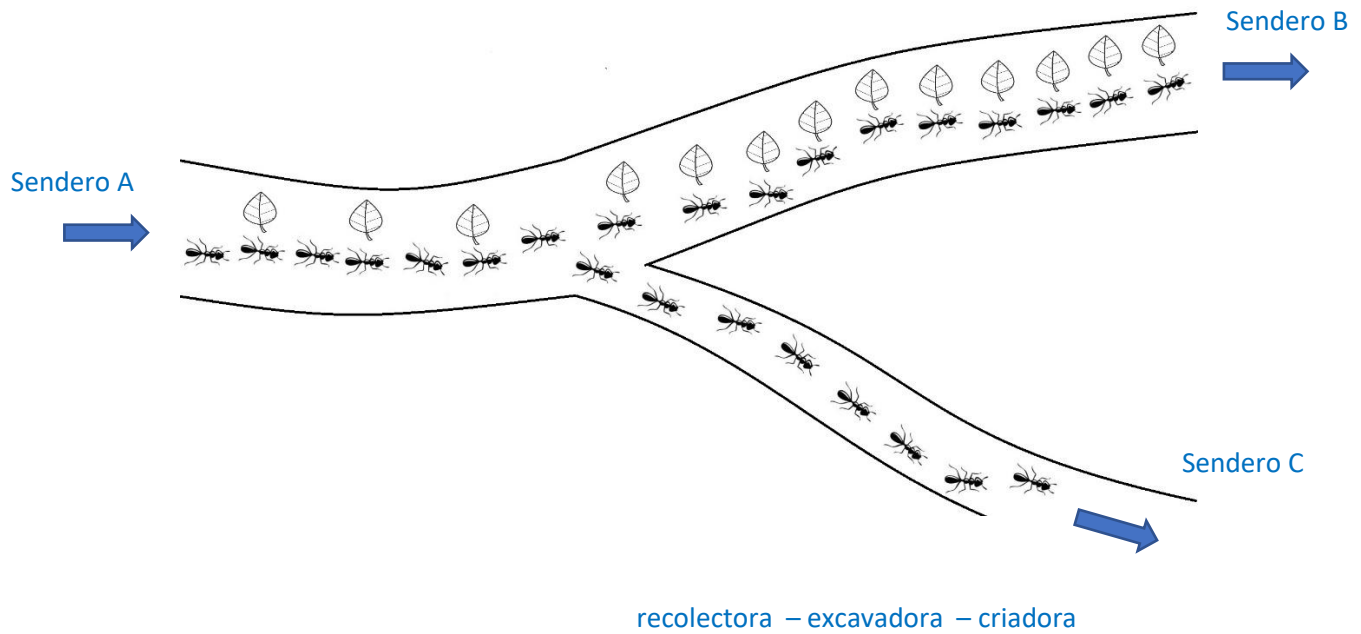


**EJERCICIO 1:** Una fila de hormigas africanas, esta recolectando hojas para la supervivencia de la colonia. Las mismas se mueven en un sendero marcado por rastros químicos olorosos que marcan un camino y que todas ellas siguen para llegar, por ejemplo, a una fuente de alimento. Por tanto, una hormiga no puede adelantar a otra que se ha formado mas adelante.



Realice la struct *Hormiga* que puede ser de uno de 3 tipos:

Guarda tipo, tamaño, y si recolectó o no una hoja (V/F)



Vamos a crear el *senderoA*, la FILA de todas las hormigas que tienen rumbo a sus tareas.

Todas las hormigas recolectoras, y que además tengan hojas, se insertan al *senderoB*(FILA), de lo contrario si no llevan hoja van a la FILA *senderoC*. Luego de que depositan la hoja salen del *senderoB*(suprimir), y regresan al *senderoA* principal, seteando previamente la hoja en *false*.

Hacer un menu de opciones, usando do-while y switch para:

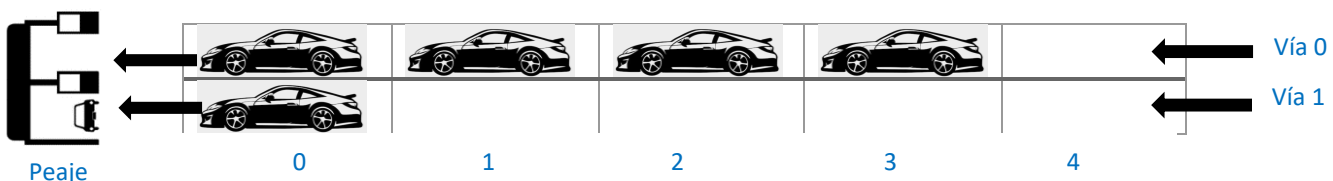
1. Crear hormigas
2. Cargar Hormiga a sendero A
3. Derivar a sendero B o C
4. Quitar hoja, Regresar de sendero B a A
5. Mostrar y Vaciar sendero C

## EJERCICIO 2:

Se cuenta con una autopista con peaje de 2 carriles: Via0 (común) y Via1(la vía rápida). Simule un TDA peaje, que se adapta del TDA FILA DUO, para lo cual tendremos métodos que permitan (*insertarV0/insertarV1*) y (*suprimirV0/suprimirV1*) en TDA peaje, los vehículos en ambas vías.



Para ello debemos contar con un struct **TDApeaje** que se implementa con una **matriz de 2x5** (2 vías de 5 cupos como máximo). Los elementos de la **matriz** son de tipo **struct auto**, que guarda modelo y patente. Cargar, por ejemplo: "Vento", "AZ 234 RT" en via1, y "Cronos", AS 459 GT en via2.



Como usuario de TDA Peaje, en el main, cargue 4 vehículos en la vía común (fila 0), y 1 en la rápida. Luego desencole un vehículo de cada vía, y muestre los resultados.

**A- Cadenas:**

Declare una variable `palabra1` con valor inicial 'palabra' (**strcpy**), y en el programa que permita ingresar otra parte de la `palabra1`, con **getline**, ej. ' Uno Dos' y anexarla (**strcat**).

Resultado -> `palabra1` queda con: palabra Uno Dos

Luego tendrá opciones para:

- a. Determinar la longitud de la cadena (**strlen**)
- b. Solicitar otra cadena, y
  - i. Verificar si es una subcadena de la primera (**strstr**)
  - ii. Anexarla a la primera cadena separado por un guion, y comparar cuál de las dos es la cadena más larga (**strcat**) (**strlen**)
- c. Solicitar otra letra, y verificar si es esta en la primera cadena (**strchr**)
- d. Cargar en una variable cadena la frase "Examen de EDA", y determinar si es igual a la cadena ingresada por teclado, independientemente de las mayúsculas. (**strcasecmp**)

**B- Bolsa de trabajo.** Escribir las siguientes estructuras anidadas:

- *Persona* con al menos 20 caracteres para el nombre, edad, peso y la fecha de nacimiento, que debe ser una *Fecha*
  - *Fecha* es una estructura compuesta de su día, mes y año.
1. Pedir en el main, los datos de la persona y precargar a Verónica, edad 20, peso 45, nacida el 2 de diciembre de 1997.
    - Escribir la estructura *Empleado* que es, una persona con un salario y fecha de contratación.
  2. Desde el main creamos a Esteban de 19 años que pesa 60kg, y cumple años el 22 de julio de 1998. Esteban cuenta con un salario de \$ 351.667.19.
  3. Desde el main, nos ocuparemos de mostrar tanto a Esteban como a Verónica.
  4. Declarar un arreglo de trabajadores, y cargar a Esteban y dos trabajadores adicionales, luego mostrar todos sus campos