Sintaxis y Semántica de los Lenguajes

Curso K2054

TRABAJO PRÁCTICO

AUTÓMATAS

**Docente** - Roxana Leituz

| Fecha de entrega: | 06/11/2024 | Nota: |  | Fecha Aprobación: |  |
| --- | --- | --- | --- | --- | --- |

| **Grupo: 03** |  |  |
| --- | --- | --- |
| **Cimino**, Facundo | 213.601-6 | fcimino@frba.utn.edu.ar |
| **Astulli de la Cruz**, Cristian | 213.262-0 | castullidelacruz@frba.utn.edu.ar |
| **Kibel**, Ian | 213.811-6 | ikibel@frba.utn.edu.ar |

| Comentarios: |
| --- |

**Trabajo Práctica Autómatas**

* Todos los trabajos deben llevar carátula con todos los datos formales.
* Se deben incluir impresiones de las pantallas mostrando el funcionamiento del mismo.
* La entrega debe contener los archivos fuente y un pdf con las pantallas, instructivo y especificaciones acerca de las decisiones tomadas para la resolución del mismo.

**Actividades**

1. Dada una cadena que contenga varios números que pueden ser decimales, octales o hexadecimales, con o sin signo para el caso de los decimales, separados por el carácter ‘#’, reconocer los tres grupos de constantes enteras, indicando si hubo un error léxico , en caso de ser correcto contar la cantidad de cada grupo.

* Debe diagramar y entregar el o los autómatas utilizados y las matrices de transición.
* La cadena debe ingresar por línea de comando o por archivo.

1. Debe realizar una función que reciba un carácter numérico y retorne un número entero.
2. Ingresar una cadena que represente una operación simple con enteros decimales y obtener su resultado, se debe operar con +,\_ ,/ y \*. Ejemplo = 3+4\*7+3-8/4 = 32. Debe poder operar con cualquier número de operandos y operadores respetando la precedencia de los operadores aritméticos.

* La cadena ingresada debe ser validada previamente preferentemente utilizando las funciones del ejercicio 1.
* Para poder realizar la operación los caracteres deben convertirse a números utilizando la función 2. La cadena debe ingresar por línea de comando o por archivo.

**NOTA**: Si consideran que requieren buscar alguna función en la web que los auxilie con la resolución de la precedencia de operadores, pueden hacerlo. Deben citar la fuente, e incorporar una explicación del procedimiento, la copia de la web es solo a este efecto, nada del trabajo puede ser copiado, debe ser original. Si se detecta copia automáticamente pierden la promoción y tendrán que hacer un trabajo extra para aprobar la materia.

**Resolución**

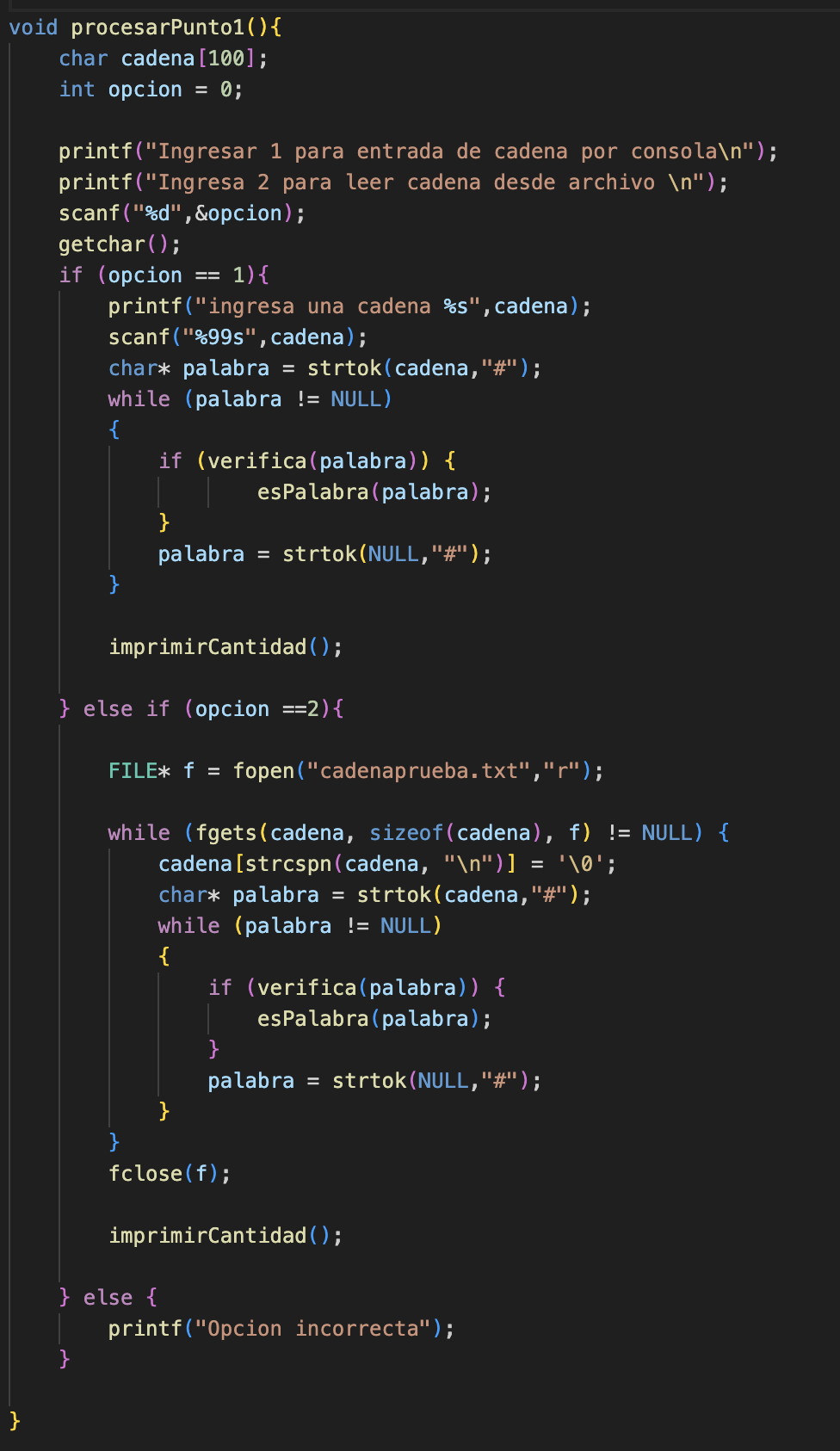
**Ejercicio 1 - Procesar cadenas**

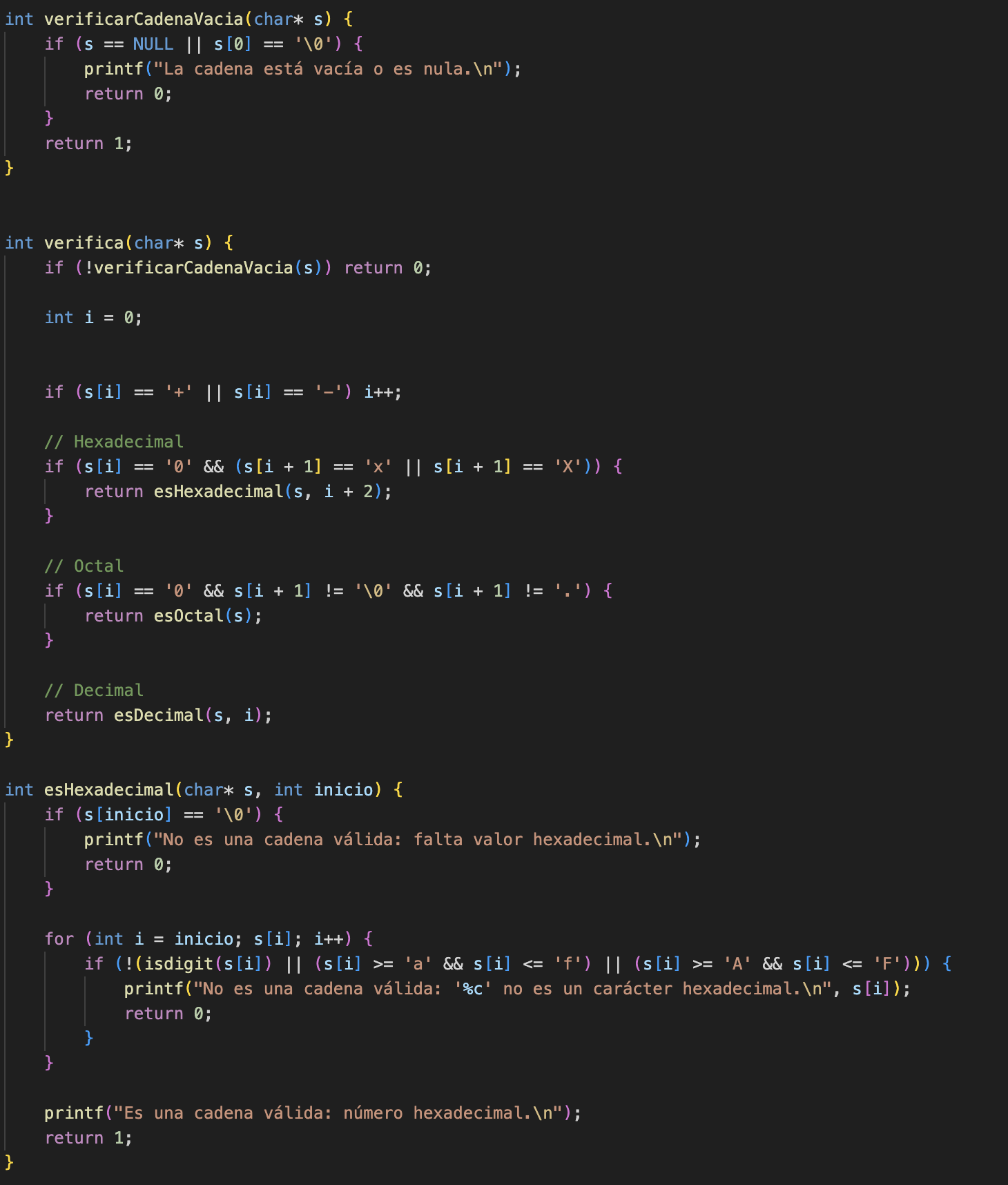
Este ejercicio tiene como objetivo procesar una cadena de texto y clasificar los números según su base (decimal, octal o hexadecimal).

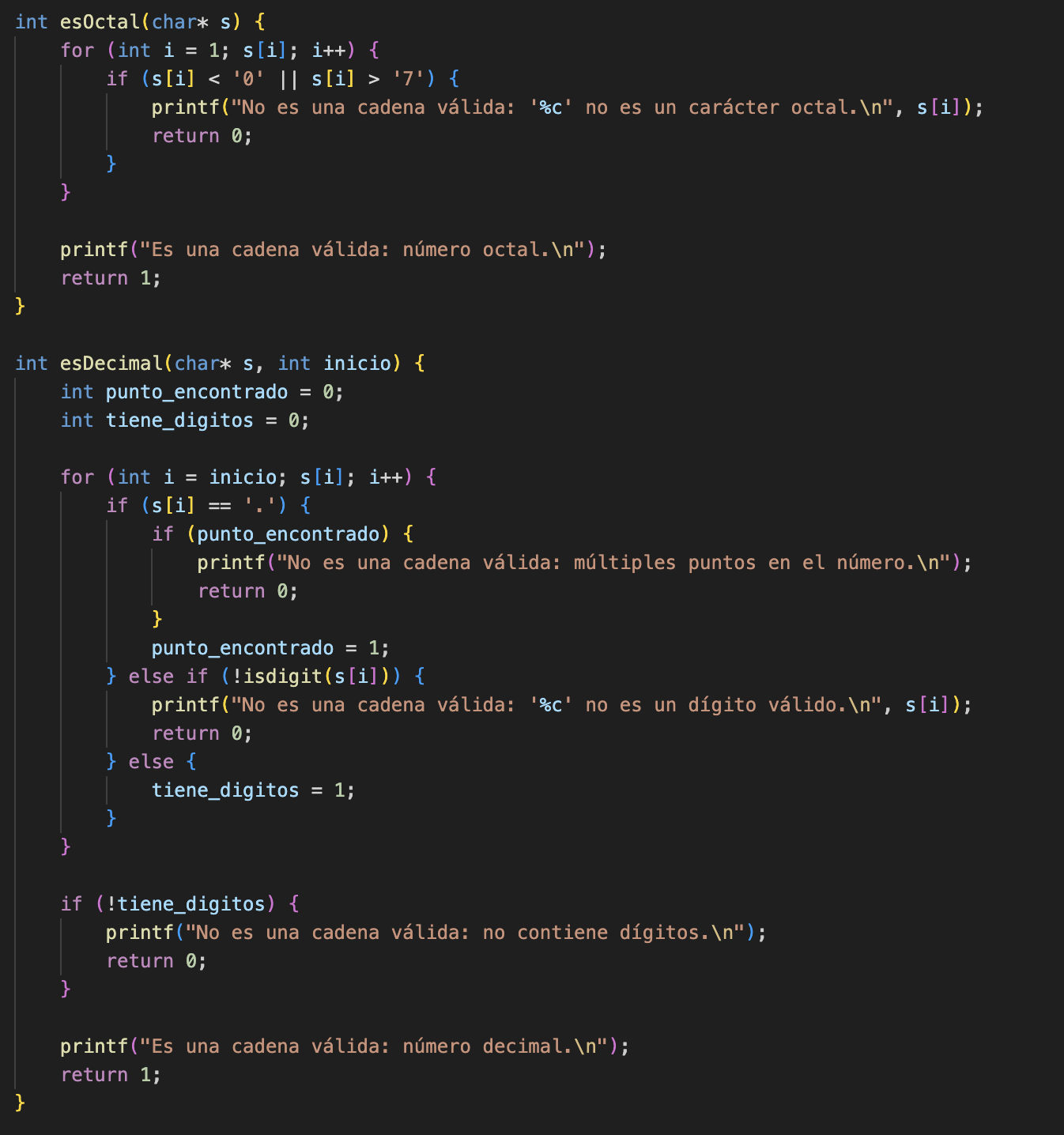
El programa sigue los siguientes pasos para procesar las cadenas:

1. El programa pide al usuario ingresar una cadena de texto.
2. La cadena se procesa utilizando *strtok*() para separar las palabras y luego se valida si son números válidos aplicando las reglas de formato para decimales, octales y hexadecimales.
3. Se utiliza la función *verifica*() para validar la cadena y clasificarla como decimal, octal o hexadecimal.
4. Finalmente, se imprimen las cantidades de números de cada tipo encontrados.

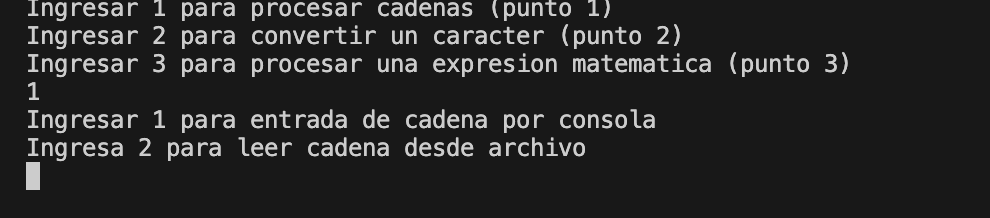
El fragmento de código relevante para el desarrollo de este ejercicio, se concentra en las funciones *procesarPunto1()* y *verifica()*. Para visualizar el código completo, vea el archivo llamado “*tpAutomatas.c*” ubicado en la carpeta “*src*” del repositorio de la entrega.



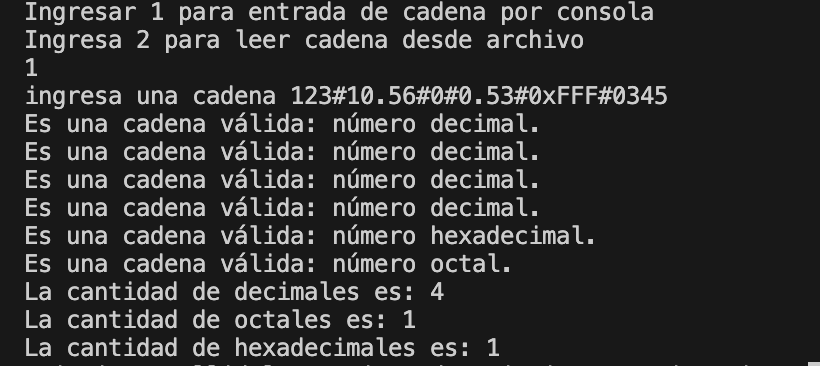




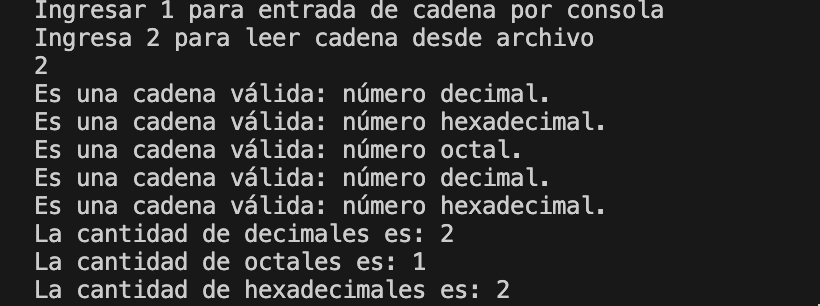
Al ingresar “1”, se genera un nuevo menú que permite elegir la modalidad de ingreso (consola/archivo), en caso de ser por consola se muestra una nueva pantalla para ingresar las cadenas separadas por “#”.



Cadena ingresada por consola:



Cadena ingresada por archivo:



En este ejercicio, se procesan las cadenas que contienen números en diferentes bases. La función *verifica*() comprueba si el número es válido y está clasificado correctamente.

Las expresiones válidas pueden ser de la forma:

Y las tablas de transición se verían de la siguiente manera:

|  | - | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0- | 1 | 5 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 9 | 9 |
| 1+ | 9 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 9 | 9 |
| 2+ | 9 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 9 | 9 |
| 3 | 9 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 9 | 9 |
| 4+ | 9 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 9 | 9 |
| 5+ | 9 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 9 | 9 | 9 | 9 |
| 6+ | 9 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 9 | 9 | 9 | 9 |
| 7 | 9 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| 8+ | 9 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| 9 (r) | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |
| 10 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |

| c | d | e | f | A | B | C | D | E | F | , | x | X |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |
| 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 3 | 9 | 9 |
| 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 3 | 9 | 9 |
| 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |
| 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |
| 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 7 | 7 |
| 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |
| 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 9 | 9 | 9 |
| 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 9 | 9 | 9 |
| 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |
| 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |

**Ejercicio 2**

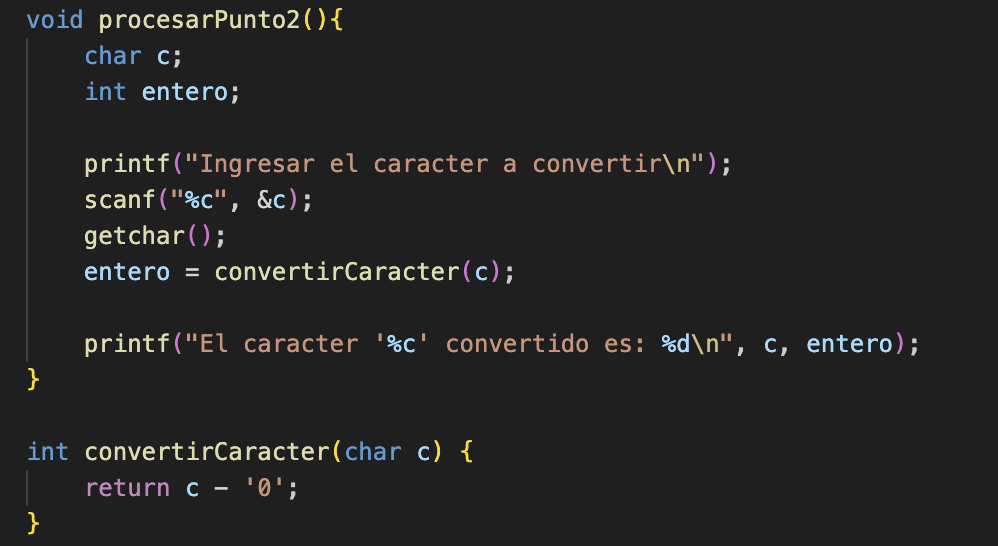
**Conversión de caracteres**

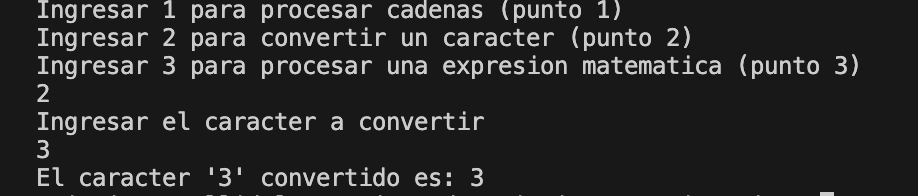
Este ejercicio tiene como objetivo crear una función que reciba un carácter numérico (por ejemplo, '0', '1', '2', ...) y retorne el valor numérico entero correspondiente.

El programa sigue los siguientes pasos para procesar las operaciones matemáticas:

1. La función *procesarPunto2*() toma un carácter numérico (como '3') y lo convierte a su valor entero correspondiente (en este caso, 3) utilizando la función convertirCaracter().
2. La función *procesarPunto2*() muestra el resultado de la conversión en la consola, llamando a *convertirCaracter*() y luego imprimiendo el valor entero.

El fragmento de código relevante para el desarrollo de este ejercicio, se concentra en la función *procesarPunto2*() y *convertirCaracter()*. Para visualizar el código completo, vea el archivo llamado “*tpAutomatas.c*” ubicado en la carpeta “*src*” del repositorio de la entrega.





La función *convertirCaracter*() convierte el carácter numérico en un valor entero, utilizando la diferencia entre el valor ASCII del carácter y el valor ASCII del '0'. Esto da como resultado el valor numérico correspondiente.

**Ejercicio 3 - Evaluación de expresiones matemáticas**

Este ejercicio tiene la función de evaluar expresiones matemáticas con operadores básicos (+, -, \*, /). Se hace mediante una función que analiza una expresión ingresada como cadena y la evalúa utilizando dos pilas: una para operandos y otra para operadores.

El programa sigue los siguientes pasos:

1. El programa lee un archivo de texto que contiene expresiones matemáticas.
2. Las expresiones se procesan y se evalúan de manera similar al ejercicio anterior, pero ahora se leen desde el archivo en lugar de ingresarlas por consola.

El fragmento de código relevante para el desarrollo de este ejercicio, se concentra en las funciones *procesarPunto3(), evaluarExpresion(), esOperador(), precedencia(), y realizarOperacion()*. Para visualizar el código completo, vea el archivo llamado “*tpAutomatas.c*”.

La función esOperador() se encarga de determinar si un caracter es operador matemático, o no, devolviendo 1 en caso afirmativo.

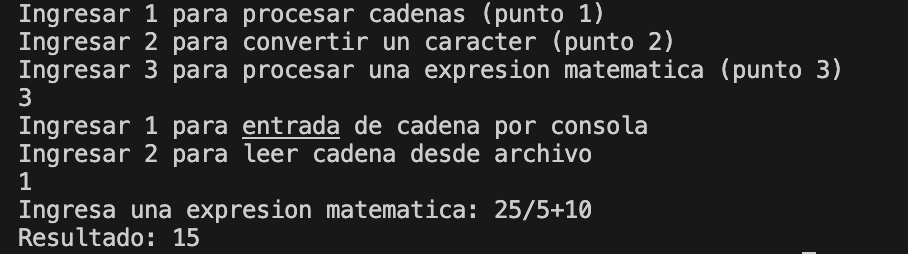
Luego, es necesario definir la función precedencia, la cual le da prioridad alta a la multiplicación y división.

Es necesario para el funcionamiento, definir la función que realice la operación, la cual la hace en base a dos operandos, y al operador dado.

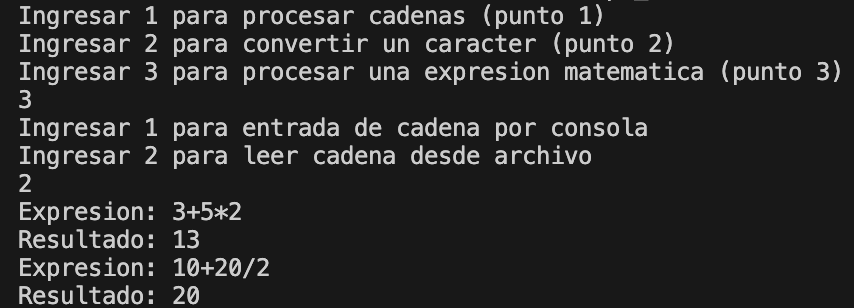
Por último, se lleva a cabo la función evaluarExpresion, la cual crea dos pilas de operandos y operadores, en las cuales se almacenan los números y los operadores correspondientes.



Por consola:



Por archivo:



En este ejercicio, se lee la expresión, se procesa usando pilas y se devuelve el resultado. Es modificable para agregar más operadores o manejar errores específicos.