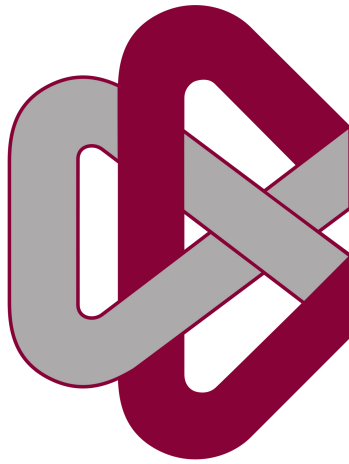


Parte 1: Lenguaje C

Tarea 5



CIMAT

Name:	Jairo Saul Diaz Soto
Maestría:	Ciencias Computacionales
Modulo:	Programación y algoritmos I
Instructor:	Dr. Rogelio Hasimoto Beltran
Fecha de entrega:	2023 08 17

1. **Union.** Este es un tipo de dato, derivado, el cual permite almacenar uno o más tipos de datos, primitivos o derivados, en un mismo bloque de memoria.

La memoria que se va a ocupar, es al menos el valor de bytes necesarios para almacenar el tipo de dato que más espacio necesite dentro de los miembros de la unión.

Dado un momento, solamente un miembro y por ende, un tipo de dato, puede ser referenciado a la vez por una unión.

La declaración de este tipo de dato se hace mediante la palabra reservada ***union*** seguido por el nombre que se desee especificar para esta union. A continuación un ejemplo

```
union ej_union{
    int entero;
    float flotante;
}
```

Listing 1: Ejemplo de tipo de dato *union*.

Para el ejemplo anterior, la unión *ej_union*, puede tomar el tipo de dato entero o flotante, pero es almacenado en un bloque de tipo flotante, es decir, un bloque contiguo (comúnmente) de 32 bytes.

Para la declaración de este nuevo tipo de dato, se realiza como ya es costumbre dentro del lenguaje de programación C, es decir, comenzamos por indicar el tipo de dato, seguido del nombre que queremos dar a esta variable. Es posible inicializar los *union*, sin embargo, estos sólo se pueden inicializar con el tipo de dato que es colocado como primer miembro en la declaración de la unión, es decir, para nuestro caso sólo sería válida la inicialización como

```
union ej_union mi_union1 = {10}; //Inicializacion correcta
union ej_union mi_union2 = {1.0}; //Inicializacion incorrecta
```

Listing 2: Inicialización de una union.

esto debido a que el primer miembro de la union es un tipo entero mientras que el segundo es un tipo flotante.

Si ahora, nosotros asignamos un valor a la parte entera de nuestra unión, nuestro tipo de dato puede ser referenciado como entero o como flotante, lo mismo aplicaría para el caso de asignar un valor al miembro flotante, pero si asignamos alguno de los dos tipos, se debería referenciar por su ultima asignación, ya que la representación sería complicada de tomar como uno u otro tipo de dato.

```
union ej_union mi_union;
mi_union.entero = 100;
printf("Referenciando la union como entero, el valor es: %d\t%f\n",
    mi_union.entero, mi_union.flotante);
mi_union.flotante = 100.0;
printf("Referenciando la union como flotante, el valor es: %d\t%f\n",
    mi_union.entero, mi_union.flotante);
```

Listing 3: Referenciar la union segun su ultima declaracion.

Donde la salida de este fragmento de código sería

```
Referenciando la union como entero, el valor es 100      0.000000
Referenciando la union como flotante, el valor es 17096   100.000000
```

Es responsabilidad del programador hacer el uso correcto de la referencia para la union.

2. **enum**. Este es otro tipo de dato que se permite manejar en C, el cual permite al programador crear una lista de constantes enteras representadas por identificadores. Es decir es una lista de constantes simbólicas.

Por defecto, esta lista comienza en 0, sin embargo, el programador puede indicar el punto de inicio y la lista se seguirá hasta donde se proponga incrementándose sucesivamente en uno.

La declaración se llevará a cabo utilizando la palabra reservada **enum** y el nombre que se desee asignar a esta enumeración, a continuación un ejemplo

```
enum num_romanos0 {I, II, III, IV, V, VI, VII, VIII, IX, X};
enum num_romanos1 {I=1, II, III, IV, V, VI, VII, VIII, IX, X};
```

Listing 4: Ejemplos de enumeracion.

Para lo anterior, entonces *num_romanos0*, tendrán valores correspondientes del 0 al 9, mientras que para el caso *num_romanos1*, como se realiza la especificación de comenzar en 1, estos serán del 1 al 10.

Ahora, la declaración de estos, nuevamente se realiza a través de la especificación del tipo de dato, seguido del nombre de la variable y este puede ser inicializado mediante algún identificador, como se muestra en el ejemplo siguiente

```
//asigna el valor 5 a la variable "cinco"
enum num_romanos1 cinco = V;
```

Listing 5: Declaración y asignación de un enum.

Cabe destacar, que tambien es posible el asignarle cualquier valor a cada uno de las constantes simbolicas declaradas en el tipo de dato, realizando la especificación para cada una, las cuales inclusive pueden tomar el mismo valor.