



**Universidad Nacional
Autónoma de México**

Facultad de Ingeniería



Estructuras de Datos y Algoritmos I

Actividad #2: Acordeón del Lenguaje C y C#

Alumno: Camacho Garduño Miguel Angel

Fecha: 26/02/2021

Lenguaje C

- `- ~ !` Operadores de negación y complemento
- `#define`: hace que el compilador sustituya por la cadena de token para cada aparición del identificador en el archivo de código fuente.
- `#include`: inserta contenido o funciones de otro archivo o librería
- `& | ^` Operadores bit a bit
- `&& ||` Operadores lógicos
- `* &` Operadores de direccionamiento indirecto y address-of
- `* / %` Operadores de multiplicación
- `,` Operador de evaluación secuencial
- `/*...*/` o `//`: Texto que omite el compilador y que sirve para documentar el código
- `+ -` Operadores aditivos
- `+` Operador unario más
- `++ --` Operadores unarios de incremento y decremento
- `< > <= >= == !=` Operadores relacionales
- `<< >>` Operadores de desplazamiento
- **Array**: es una colección de elementos de datos similares almacenados en ubicaciones de memoria contiguas y se puede acceder a los elementos de forma aleatoria utilizando índices de una matriz. Se pueden utilizar para almacenar colecciones de tipos de datos primitivos como `int`, `float`, `double`, `char`, etc. de cualquier tipo en particular.
- **Break**: se usa con frecuencia para finalizar el procesamiento de un caso concreto en una instrucción `switch`.
- **Char**: Almacena un solo carácter y requiere un solo byte de memoria en casi todos los compiladores
- **Default**: se ejecuta si ningún valor `constant-expression` case es igual al valor de `expression`. Si no hay ninguna instrucción `default` y no se encuentra ninguna instrucción case coincidente, no se ejecuta ninguna de las instrucciones del cuerpo `switch`
- **Do**: permite repetir una instrucción o una instrucción compuesta hasta que una expresión especificada sea `false`.
- **Double**: se utiliza para almacenar números decimales (números con valor de coma flotante) con doble precisión.
- **Else**: se ejecuta la segunda instancia de `statement` si `expression` es `false`

- Float: variable de tipo flotante
- For: permite repetir una instrucción o una instrucción compuesta un número especificado de veces. El cuerpo de una instrucción for se ejecuta cero o más veces hasta que una condición opcional sea false
- If: controla la bifurcación condicional. El cuerpo de una instrucción if se ejecuta si el valor de la expresión es distinto de cero
- Int: variable de tipo entero
- Long: un tipo de variable con un intervalo de valores de -2.147.483.648 a 2.147.483.647
- Main(): sirve como punto de partida para la ejecución del programa. Normalmente, controla la ejecución del programa dirigiendo las llamadas a otras funciones del programa
- Printf(""): escribe en la entrada estándar
- Return: termina la ejecución de una función y devuelve el control a la función de llamada. La ejecución se reanuda en la función de llamada, en el punto que sigue inmediatamente a la llamada. Una instrucción return puede devolver un valor a la función de llamada
- Scanf: se usa para recibir información del usuario.
- Short: un tipo de variable con un intervalo de valores de -32 768 a 32 767
- Signed: el tipo de variable por defecto
- sizeof Operador de tamaño
- Static: Una variable que conserva su valor entre múltiples llamadas a funciones se conoce como variable estática
- Switch-Case: ayudan a controlar las operaciones condicionales y de bifurcación complejas. La instrucción switch transfiere el control a una instrucción dentro del cuerpo
- Unsigned int: un tipo de variable con un intervalo de variables de 0 a 4.294.967.295
- Void: Se utiliza para definir una función que no devuelve ningún valor o declarar punteros genéricos.
- While: permite repetir una instrucción hasta que una expresión especificada sea false

C#

- #define: hace que el compilador sustituya por la cadena de token para cada aparición del identificador en el archivo de código fuente.
- #include: inserta contenido o funciones de otro archivo o librería

- **Array:** es una colección de elementos de datos similares almacenados en ubicaciones de memoria contiguas y se puede acceder a los elementos de forma aleatoria utilizando índices de una matriz. Se pueden utilizar para almacenar colecciones de tipos de datos primitivos como int, float, double, char, etc. de cualquier tipo en particular.
- **Break:** se usa con frecuencia para finalizar el procesamiento de un caso concreto en una instrucción switch.
- **c ? t : f** Operador condicional
- **Char:** Almacena un solo carácter y requiere un solo byte de memoria en casi todos los compiladores
- **Default:** se ejecuta si ningún valor constant-expression case es igual al valor de expression. Si no hay ninguna instrucción default y no se encuentra ninguna instrucción case coincidente, no se ejecuta ninguna de las instrucciones del cuerpo switch
- **Do:** permite repetir una instrucción o una instrucción compuesta hasta que una expresión especificada sea false.
- **Double:** se utiliza para almacenar números decimales (números con valor de coma flotante) con doble precisión.
- **Else:** se ejecuta la segunda instancia de statement si expression es false
- **Float:** variable de tipo flotante
- **For:** permite repetir una instrucción o una instrucción compuesta un número especificado de veces. El cuerpo de una instrucción for se ejecuta cero o más veces hasta que una condición opcional sea false
- **If:** controla la bifurcación condicional. El cuerpo de una instrucción if se ejecuta si el valor de la expresión es distinto de cero
- **Int:** variable de tipo entero
- **Long:** un tipo de variable con un intervalo de valores de -9.223.372.036.854.775.808 a 9.223.372.036.854.775.807
- **Main():** sirve como punto de partida para la ejecución del programa. Normalmente, controla la ejecución del programa dirigiendo las llamadas a otras funciones del programa
- **Printf(""):** escribe en la entrada estándar
- **Return:** termina la ejecución de una función y devuelve el control a la función de llamada. La ejecución se reanuda en la función de llamada, en el punto que sigue inmediatamente a la llamada. Una instrucción return puede devolver un valor a la función de llamada
- **Scanf:** se usa para recibir información del usuario.

- Short: un tipo de variable con un intervalo de valores de -32 768 a 32 767
- Signed: el tipo de variable por defecto
- sizeof Operador de tamaño
- Static: Una variable que conserva su valor entre múltiples llamadas a funciones se conoce como variable estática
- Switch-Case: ayudan a controlar las operaciones condicionales y de bifurcación complejas. La instrucción switch transfiere el control a una instrucción dentro del cuerpo
- Unsigned int: un tipo de variable con un intervalo de variables de 0 a 4.294.967.295
- Void: Se utiliza para definir una función que no devuelve ningún valor o declarar punteros genéricos.
- While: permite repetir una instrucción hasta que una expresión especificada sea false
- x & y Operador lógico booleano AND u operador lógico bit a bit AND
- x && y AND condicional
- x * y, x / y, x % y Multiplicativo
- x ?? y Operador de uso combinado de null
- x ^ y Operador lógico booleano XOR u operador lógico bit a bit XOR
- x | y Operador lógico booleano OR u operador lógico bit a bit OR
- x || y OR condicional
- x + y, x - y Aditivo
- x < y, x > y, x <= y, x >= y, is, as Comprobación de tipos y relacional
- x << y, x >> y Shift
- x = y, x += y, x -= y, x *= y, x /= y, x %= y, x &= y, x |= y, x ^= y, x <= y, x >= y, x ??= y, => Asignación y declaración lambda
- x == y, x != y Igualdad

Bibliografía

GeeksforGeeks. (3 de noviembre de 2020). *Data Types in C*. Recuperado el 26 de febrero de 2021, de GeeksforGeeks: <https://www.geeksforgeeks.org/data-types-in-c/>

Microsoft Corporation. (s.f.). *Documentos de C#: inicio, tutoriales y referencias*. Recuperado el 26 de febrero de 2021, de Microsoft Docs: <https://docs.microsoft.com/es-mx/dotnet/csharp/>

Microsoft Corporation. (s.f.). *Documentos de C: inicio, tutoriales y referencias*. Recuperado el 26 de febrero de 2021, de Microsoft Docs: <https://docs.microsoft.com/es-mx/cpp/c-language/?view=msvc-160>