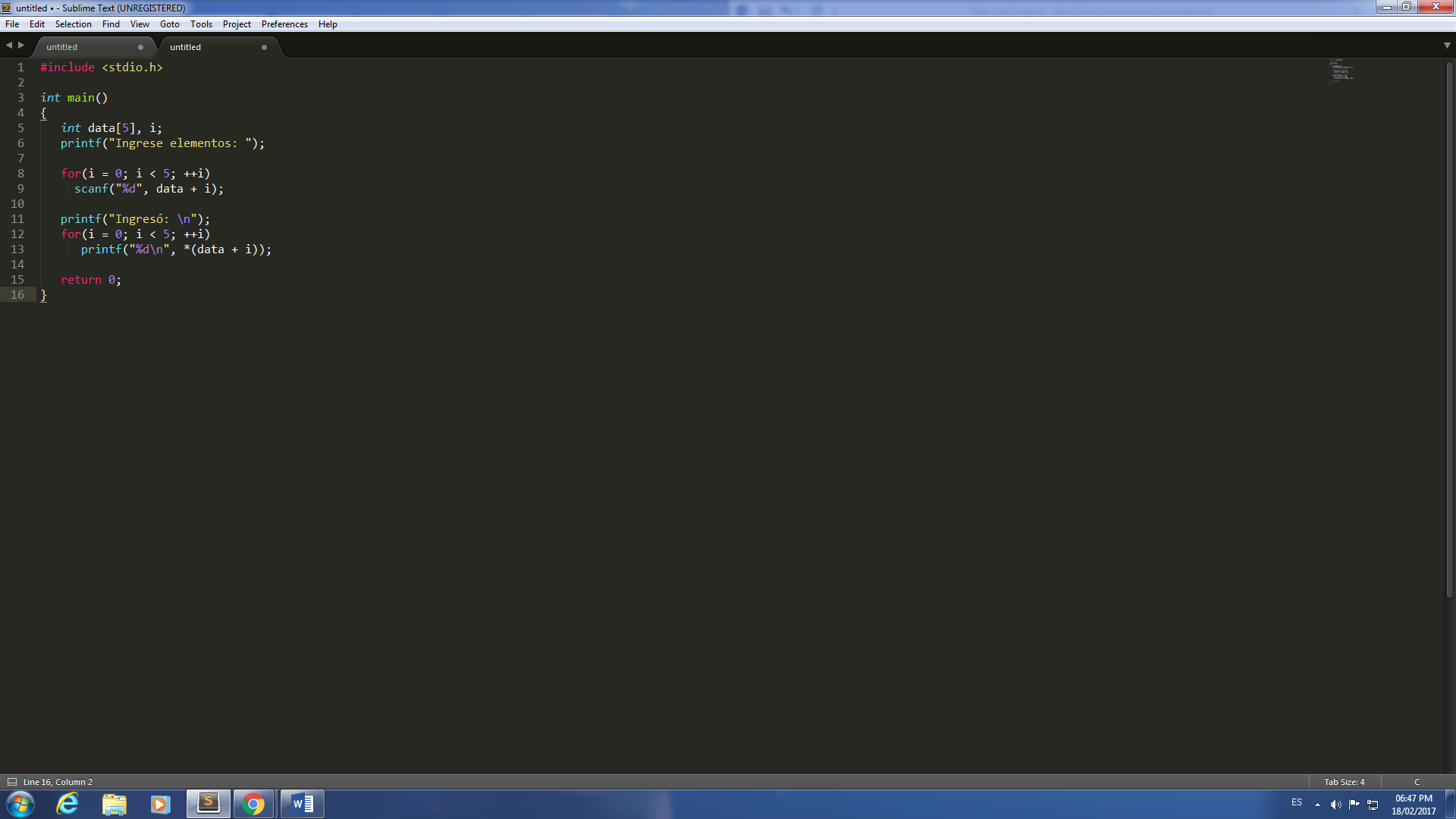
Jonathan Asprilla Saavedra

1. Es llamado un lenguaje de nivel medio porque se puede manejar la memoria y su sintaxis y semántica es cercana al lenguaje natural.
2. Características:
   1. En las líneas de código se usa el ***;*** , exceptuando los ***while*** y los ***if.***
   2. Hay que declarar el tipo de las variables a usar.
   3. Se compila usando la terminal.
   4. Es simple y portable.
   5. Usa punteros y maneja la memoria.
3. Las variables locales se usan nomás en la función donde las declaras. En cambio, las variables globales se pueden usar en todo el programa.
4. Un arreglo en C, es un conjunto de variables del mismo tipo.
5. Un puntero en C, es una variable que contiene la dirección de memoria de un dato.
6. Los punteros son peligrosos en malas manos porque podrían re-escribir la memoria del sistema.
7. Misma respuesta que la 6.
8. Sí, porque con los punteros podemos referirnos a la dirección de memoria en la que se encuentra el arreglo.
9. “Void” se usa en funciones que no retornan nada.
10. Ofrecen potencia y flexibilidad en C. Se pueden tener apuntadores a cualquier tipo de variable. También son útiles para pasarle parámetros a las funciones de tal modo que les permiten modificar y regresar valores a la rutina que las llama.
11. Prácticamente, break se usa en switch, cuando queremos salir después de que un caso particular es ejecutado; en bucles, cuando queremos abandonar un bucle tan pronto como se produzca una determinada condición (por ejemplo, se detecta una condición de error o se llega al final de los datos prematuramente).
12. La diferencia entre “Char a” y “Char a[1]” es que, el primero se refiere a que la variable **a** es de tipo carácter, mientras que el segundo se refiere a que en el índice **1** del arreglo **a** se encuentra un elemento de tipo carácter.
13. Las cadenas son inmutables mientras que las matrices no lo son. Las matrices se almacenan en bloques de memoria continua en la memoria, pero las cadenas se pueden almacenar de cualquier manera.
14. Un compilador es un programa informático que traduce un programa escrito en un lenguaje de programación a otro lenguaje diferente.
15. Con el modificador **signed** o usando simplemente el **int**.
16. En programación, si una estructura en el heap (lugar de donde se asigna memoria) es destruida antes de que todas las referencias (los apuntadores) a la misma sean destruidas, cualquier referencia restante se llama referencia colgante (Dangling Reference).
17. Las variables locales se declaran por dentro de las funciones para ser usadas solo cuando se realice el llamado a la función.
18. No, porque una variable **long int**, sobrepasa la memoria de una variable de tipo **float**.
19. Los operadores relacionales retornan con un 0 la condición de **false**, y cualquier número distinto de 0 equivale a la condición **true**.
20. El valor de las variables es, como su propio nombre indica, variable. Podemos alterar su valor en cualquier punto del programa.
21. Si se puede acceder a un arreglo por medio de un puntero, un ejemplo de código seria:



1. La longitud de una cadena es n -1 debido a que se define como el número de símbolos que la componen, sin contar el terminador de cadena, siendo el terminador de la cadena el símbolo ´\0´.
2. Así como las variables, las funciones también deben ser declaradas. La correcta manera de declarar funciones es:



1. La principal diferencia se centra en la manera en que se ejecutan los lenguajes de programación. Por un lado el lenguaje Python es un lenguaje interpretado el cual es leído línea por línea y comúnmente no se guarda el resultado de dicha traducción. Al contrario el lenguaje C, es un lenguaje compilado donde se crea código de maquina a partir de código fuente.
2. Se conoce a C como el lenguaje de nivel medio porque se puede manejar la memoria y su sintaxis y semántica es cercana al lenguaje natural.
3. Los lenguajes de alto nivel permiten al programador utilizar palabra muy cercanas al lenguaje del común, por ende es considerado mucho más fácil de aprender y así mismo ejecutar. Los lenguajes intermedio o de medio nivel son aquellos que no crean código objeto o código maquina por si solos sino que solo facilitan el proceso de optimización. Al final, los lenguajes de bajo nivel son aquellos que con sus características ejercen un control directo sobre el hardware y están condicionados con la estructura del computador donde se ejecute.

|  |  |  |
| --- | --- | --- |
| ***Alto nivel*** | ***Medio nivel*** | ***Bajo Nivel*** |
| C++ | C | Ensamblador - Assembly |

1. Los compiladores son programas informáticos que traducen los programas escritos en lenguajes de programación a lenguajes diferentes. Normalmente todo es traducido al lenguaje máquina pero también puede ser cod intermedio o texto.

|  |  |  |
| --- | --- | --- |
| ***Ensambladores*** | ***Compiladores*** | ***Interpretes*** |
| Traductor para lenguajes de bajo nivel. | Traductor para lenguajes de alto nivel. | Traductor parecido al compilador pero sin creación de un ejecutable. |

1. Se inicia con la función main ().
2. El lenguaje C es sensible a mayúsculas y minúsculas debido a que trabaja con el código ASCII y por ende cada letra tiene un correspondiente valor que la hace diferente de las demás.

|  |  |  |  |
| --- | --- | --- | --- |
| ***INT*** | ***CHAR*** | ***FLOAT*** | ***DOBLES*** |
| Tipo de datos de valores numéricos enteros. | Tipo de datos de valores alfabéticos. Un solo carácter. | Tipo de datos de valores numéricos decimales. | Tipo de datos que admiten rangos más amplios en memoria. |

2. La función sizeof() en C , devuelve el tamaño en bytes que ocupa una variable o algún tipo de dato.
3. Los diferentes tipos son:

* Short.
* Long.
* Signed.
* Unsigned.

1. Una variable local es aquella que se restringe a ser usada donde se le fue declarada.
2. Una variable global es aquella que se define fuera del cuerpo de una función, a las cuales cualquier función puede acceder.
3. El “=” se usa para definir valores en una variables, mientras que “==” se usa en las condiciones de que una variable llegue a ese valor que se le asigna.
4. El while hace el papel de ciclo normal, mientras que el do while ayuda a que mintras que el ciclo está cumpliendo su papel, lo que esté dentro del do se esté ejecutando.
5. El “&” se usa principalmente para la lectura de datos por teclado y el “\*” se usa para la definición de puntero antes de una variables y para definir multiplicación entre variables.
6. Podría llegar a ocurrir un ciclo infinito que puede consumir toda la memoria Ram.
7. El “#include” permite traer al programa diferentes bibliotecas que harán más fácil la programación del mismo.
8. Lo mejor es siempre definir el tamaño de un arreglo antes de usarlo debido a que se puede estar yendo hacia cualquier lugar de memoria donde ya hay otra información hasta el punto de acabar con ella.
9. Sí se puede hacer, creando una variable local que será el arreglo y otra que será leída, la cual llega a ser el tamaño del arreglo.
10. Cuando nos muestra el error de segmentación o de memoria en C, se refiere a que se quiere acceder a una posición de memoria que no le pertenece o una posición incorrecta, por ende se detiene la aplicación y se genera el error de segmentación.