

RESPUESTAS(Santiago Coca Rojas)

1. Se dice que el lenguaje C es un lenguaje de nivel medio. Esto es porque en C se pueden crear programas que manipulan la máquina casi como lo hace el lenguaje Ensamblador, pero utilizando una sintaxis que se asemeja más a los lenguajes de alto nivel.
2. Portabilidad de los programas, soporta lenguajes de bajo nivel, manipulación de bits, programación modular, y uso eficiente de punteros.
3. La única diferencia entre una variable local y una global es su ámbito de acción. Variable local: es aquella que sólo es utilizable dentro del código de la función. Variable global: es aquella cuyo ámbito es todo el programa, incluso dentro de una función.
4. Arreglo: Colección ordenada de elementos de un mismo tipo. Ordenada significa que cada elemento tiene una ubicación determinada dentro del arreglo y debemos conocerla para accederlo.
5. Un puntero es una variable que contiene la dirección de memoria de un dato o de otra variable que contiene al dato en un arreglo. Esto quiere decir, que el puntero apunta al espacio físico donde está el dato o la variable.
6. El mal uso de un puntero podría ocasionar daños en la memoria RAM del computador, lo que podría causar el colapso total del sistema.
7. El mal uso de un puntero podría ocasionar daños en la memoria RAM del computador, lo que podría causar el colapso total del sistema.
8. Si, ya que dicho apuntador guarda la dirección de memoria de dicho arreglo, lo cual facilita el acceso a este última.
9. Cuando no va a retornar nada, simplemente para ejecutar otra función que sea del tipo char.
10. Mientras un programa está corriendo todas las variables declaradas como punteros se guardan en memoria, cada una tiene su dirección o localidad única.
11. El break rompe el ciclo sin haberse ejecutado todas veces establecidas en las condiciones, al contrario del continue que ejecuta el ciclo una vez más de lo establecido.
12. Con "char a" se declara un arreglo, pero no su tamaño, en cambio con "char a[1]" se declara un arreglo que solo contendrá dos elementos, uno en la posición 0 y otro en la posición 1.
13. Los arrays son variables estructuradas, donde cada elemento se almacena de forma consecutiva en memoria. Las cadenas de caracteres son declaradas en C como arrays de caracteres y permiten la utilización de un cierto número de notaciones y de funciones especiales.
14. Programa que traduce los programas escritos en lenguajes de alto nivel a código máquina.
15. Si en un programa de ordenador se intenta asignar a un entero un valor que está fuera del rango de los valores que se pueden representar (Ej: $a=2^{40}$) se produce un fallo que se conoce con el nombre de desbordamiento (overflow)

en inglés). Cuando esto ocurre, lo habitual es que el programa siga funcionando como si nada hubiera pasado, pero el nuevo valor quedaría establecido en 0, si el desbordamiento se produce en un entero sin signo; y en -2^{31} si se produce en un entero con signo.

16. 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. Se almacenan en las pillas de llamadas.
18. Se puede asignar, pero no retorna el valor asignado originalmente.
19. No entendí la pregunta.
20. Globales=0 y a las locales inician con un valor al azar.
21. Si puede, ya que es su dirección de memoria.
22. Porque la última posición siempre está ocupada por el carácter '\0'.
23. Si no se declara una función esta no funcionaría, no sabría qué tipo de dato debe retornar y el programa no funcionaría ni compilaría.
24. Python es un lenguaje interpretado y C es un lenguaje compilado. Al ejecutarse un programa en Python, durante ese tiempo de ejecución se analiza línea por línea el código, en cambio C, al utilizar un compilador, analiza todo el código al mismo tiempo, o cual lo hace muy efectivo.
25. Se dice que el lenguaje C es un lenguaje de nivel medio. Esto es porque en C se pueden crear programas que manipulan la máquina casi como lo hace el lenguaje Ensamblador, pero utilizando una sintaxis que se asemeja más a los lenguajes de alto nivel.
26.
 - a. Alto nivel: se caracteriza por expresar el algoritmo de una manera adecuada a la capacidad cognitiva humana, en lugar de la capacidad ejecutora de las máquinas. Ej: Python.
 - b. Medio nivel: es el lenguaje de una máquina abstracta diseñada para ayudar a realizar el análisis de un programa informático. El término proviene de su uso en compiladores, donde el código fuente de un programa es traducido a un modo más apropiado para transformaciones de mejora de código antes de generar el código objeto o código máquina para una máquina determinada. Ej: C.
 - c. Bajo nivel: es aquel en el que sus instrucciones ejercen un control directo sobre el hardware y están condicionados por la estructura física de las computadoras que lo soportan. Ej: Ensamblador.
27. Programa que traduce los programas escritos en lenguajes de alto nivel a código máquina.
28. Los ensambladores son programas que procesan los enunciados del programa origen en lenguaje ensamblador y los traducen en archivos en lenguaje máquina que son ejecutados por un microprocesador o un microcontrolador. El compilador traduce el lenguaje de alto nivel a lenguaje

de máquina. El intérprete solo realiza dicha traducción si es estrictamente necesaria.

29. Empieza por la función main.
30. Si, por que relevancia escribir un carácter en mayúsculas o minúsculas.
31. Int, float and double son tipos de número, mientras que char son cadenas de texto.
32. Devuelve el tamaño en bytes que ocupa una variable o algún tipo de dato.
33. Son comandos que al acompañarlos de una variable sirven para determinar el tamaño o el signo de dicha variable.
34. Es una variable declara dentro de una función, que solo puede ser utilizada dentro de esta.
35. Es una variable declarada fuera de todas las funciones del programa, lo cual permite su uso dentro de cualquiera de las funciones que componen a dicho programa.
36. "=" se utiliza para asignar valores a variables. "==" se utiliza para realizar una comparación entre dos valores.
37. El while no se ejecuta si no se cumple la condición. Por otro lado, el do while, al tener dicha condición al final del bloque se ejecuta al menos una vez.
38. Sirven para hacer referencia a la dirección de memoria de alguna variable.
39. El programa resultaría en un ciclo infinito.
40. Se usa en los lenguajes C y C++ para "incluir" las declaraciones de otro fichero en la compilación.
41. Si, ya que si no se hace no se ejecutaría el programa.
42. Si, esto gracias a la memoria dinámica que se usa mediante los punteros.
43. Se refiere a cuando se excede en el tamaño previamente establecido para el uso de una variable, por lo general arreglos.