Punteros.

TipodeDato \*variable

-o-

TipodeDato\* variable

(cuando declaro un puntero pongo un asterisco)

Cuando pongo un & delante de una variable obtengo la dirección de memoria de la variable.

Una variable de tipo puntero guarda una dirección de memoria.

Cuando pongo un asterisco delante (SIEMPRE DELANTE, si lo pongo atrás no anda ) de una variable tipo puntero , APUNTO al DATO de la variable y no la dirección.

NULL = Direccion de memoria nula.

Int\* Puntero.

Int variableNormal.

Puedo comparar punteros con punteros.

Puntero con hexadecimales .

Puntero con un &variableNormal

Puntero =! NULL

int n[3] = {1,2,3};

int \*ptr ;

int i ;

ptr = n ;

for ( i = 0 ; i<3 ; i++)

{

printf( "%d" , \*( ptr + i ) ) ;

} ((( Lo que se hace es *ptr + i . sizeof(int) )))*

SE PUEDE TRABAJAR UN PUNTERO COMO ARRAY , O UN ARRAY COMO PUNTERO.

Si a un puntero se le carga cualquier valor, este será interpretado como si fuera:

Seleccione una:

a. un valor inicializado correspondiente al tipo de dato del puntero

b. una dirección de memoria

c. Ninguna de las anteriores es correcta

d. literalmente el valor cargado

Retroalimentación

Respuesta correcta

La respuesta correcta es: una dirección de memoria

Pregunta **2**

Incorrecta

Puntúa 0,00 sobre 1,00

Marcar pregunta

Enunciado de la pregunta

Un puntero a cadena de caracteres (string) es:

Seleccione una:

a. Todas las anteriores son correctas

b. un puntero a char

c. un puntero a un array multidimensional de char

d. un puntero a un array unidimensional de char

Retroalimentación

Respuesta incorrecta.

La respuesta correcta es: un puntero a char

Pregunta **3**

Correcta

Puntúa 1,00 sobre 1,00

Marcar pregunta

Enunciado de la pregunta

Si se tiene 'char cadena[LONGITUD]; char \*puntero;'. Entonces se puede asignar:

Seleccione una:

a. puntero = &cadena[0];

b. Las dos anteriores son correctas

c. puntero = cadena;

d. Ninguna de las anteriores es correcta

Retroalimentación

Respuesta correcta

La respuesta correcta es: Las dos anteriores son correctas

Pregunta **4**

Correcta

Puntúa 1,00 sobre 1,00

Marcar pregunta

Enunciado de la pregunta

Si se tiene 'void mostrarCadena(char\* punteroCadena){ while(\*punteroCadena != '\0'){ printf("%c",\*punteroCadena); punteroCadena++;}}'. Qué se imprime por pantalla?

Seleccione una:

a. el dato correspondiente a la dirección de memoria de cada elemento

b. Todas las anteriores son correctas

c. la dirección de memoria de cada elemento

d. el dato correspondiente a la dirección de memoria del primer elemento

e. Ninguna de las anteriores es correcta

f. la dirección de memoria del primer elemento

Retroalimentación

Respuesta correcta

La respuesta correcta es: el dato correspondiente a la dirección de memoria de cada elemento

Pregunta **5**

Correcta

Puntúa 1,00 sobre 1,00

Marcar pregunta

Enunciado de la pregunta

Si se tiene 'struct alumno {char nombre[20];int nota;}; struct alumno auxiliarAlumno; struct\* punteroAlumno; punteroAlumno = &auxiliarAlumno;'. El puntero punteroAlumno, tendrá la dirección de memoria:

Seleccione una:

a. del comienzo del grupo de campos de la estructura auxiliarAlumno

b. Ninguna de las anteriores es correcta

c. Las dos anteriores son correctas

d. de la estructura struct alumno

Retroalimentación

Respuesta correcta

La respuesta correcta es: del comienzo del grupo de campos de la estructura auxiliarAlumno

Pregunta **6**

Incorrecta

Puntúa 0,00 sobre 1,00

Marcar pregunta

Enunciado de la pregunta

Si se tiene un vector de estructuras y dicho vector se almacena en un puntero (del tipo de esa estructura), la posición de memoria que corresponde al comienzo de dicho vector. Utilizando aritmética de punteros, en cada incremento pasará a apuntar a:

Seleccione una:

a. la próxima estructura almacenada en el array

b. Las dos anteriores son correctas

c. Ninguna de las anteriores es correcta

d. la posición de memoria sumándole sizeof(tipo de esa estructura)

Retroalimentación

Respuesta incorrecta.

La respuesta correcta es: Las dos anteriores son correctas

Pregunta **7**

Incorrecta

Puntúa 0,00 sobre 1,00

Marcar pregunta

Enunciado de la pregunta

Cuando se usan los punteros a estructura la forma de acceder a un campo es:

Seleccione una:

a. Ninguna de las anteriores es correcta

b. puntero->campo

c. (\*puntero).campo

d. Las dos anteriores son correctas

Retroalimentación

Respuesta incorrecta.

La respuesta correcta es: Las dos anteriores son correctas

Pregunta **8**

Incorrecta

Puntúa 0,00 sobre 1,00

Marcar pregunta

Enunciado de la pregunta

En términos de performance, trabajar con un array de punteros es:

Seleccione una:

a. empeora la performance

b. mejora notablemente la performance

c. Ninguna de las anteriores es correcta

d. similar a trabajar con un puntero que apunta a un array

Retroalimentación

Respuesta incorrecta.

La respuesta correcta es: mejora notablemente la performance

Pregunta **9**

Correcta

Puntúa 1,00 sobre 1,00

Marcar pregunta

Enunciado de la pregunta

Un vector de punteros dentro de cada uno de sus elementos contiene:

Seleccione una:

a. un puntero

b. generalmente una direccion de memoria de otra variable

c. Las dos anteriores son correctas

d. Ninguna de las anteriores es correcta

Retroalimentación

Respuesta correcta

La respuesta correcta es: Las dos anteriores son correctas

Pregunta **10**

Incorrecta

Puntúa 0,00 sobre 1,00

Marcar pregunta

Enunciado de la pregunta

Se tiene '#define RENGLONES 5 #define COLUMNAS 10 int matriz[RENGLONES][COLUMNAS];'. Se accede a los elementos individuales del array de la siguiente manera:

Seleccione una:

a. Ninguna de las anteriores es correcta

b. matriz[renglon][columna]

c. Las dos anteriores son correctas

d. \*(\*(matriz + renglon) + columna)

Retroalimentación

Respuesta incorrecta.

La respuesta correcta es: Las dos anteriores son correctas