**Pre-Reporte**

1. ¿Qué es un apuntador?

Un Apuntador es una variable que contiene una dirección de memoria, la cual corresponderá a un dato o a una variable que contiene el dato.

Cada variable que se utiliza en una aplicación ocupa una o varias posiciones de memoria. Estas posiciones de memoria se accesan por medio de una dirección.

1. ¿Para qué pueden servir los apuntadores?

Los punteros tienen muchas utilidades; por ejemplo, nos permiten pasar argumentos (o parámetros) a una función y modificarlos. También permiten el manejo de cadenas y de arrays. Otro uso importante es que nos permiten acceder directamente a la pantalla, al teclado y a todos los componentes del ordenador. Si sólo sirvieran para almacenar direcciones de memoria no serían de mucha utilidad. Nos deben dejar también la posibilidad de acceder al contenido de esas posiciones de memoria. Para ello se usa el operador \* (operador de indirección), que no hay que confundir con el de la multiplicación.

1. ¿Cuál es la diferencia entre paso por valor y paso por referencia?

Por valor significa que la función (o subrutina) recibe sólo una copia del valor que tiene la variable, o sea que no la puede modificar. Por referencia significa que se pasa la posición de memoria donde esta guardada la variable, por lo que la función puede saber cuánto vale, pero además puede modificarla de cualquier manera.

1. ¿Para qué sirve el operador ampersand (&)?

Es una forma de saber qué direcciones ha reservado el ordenador. Se trata de usar el operador & (operador de dirección). Vamos a ver un ejemplo: definimos la variable a y obtenemos su valor y dirección.

1. ¿Para qué sirve el asterico (\*) en apuntadores?

El \* (asterisco) sirve para indicar que se trata de un puntero y debe ir justo antes del nombre de la variable, sin espacios. En la variable punt sólo se pueden guardar direcciones de memoria, no se pueden guardar datos. Vamos a volver sobre el ejemplo anterior un poco ampliado para ver cómo funciona un puntero:

|  |  |
| --- | --- |
| Código paso por valor | |
| #include <stdio.h> |  |
| int sumar(int a,int b); | Se declara la función sumar y sus parámetros |
| main (){ | Inicio de la función main |
| int num1; | Se declara num1 |
| int num2; | Se declara num2 |
| num1=5; | Se hace num1 igual a 5 |
| num2=8; | Se hace num2 igual a 8 |
| int total=sumar(num1,num2); | Se asigna el valor de la función sumar a la variable total |
| } |  |
| int sumar(int a, int b){ |  |
| int c=0; |  |
| c=(a)+(b); | En la función suma se asigna a c el valor de la suma de a y b |
| return c; | Se regresa el valor de c |
| } |  |

**Ejercicio 1:**

**Ejercicio 2:**

|  |  |
| --- | --- |
| Código paso por referencia | |
| #include <stdio.h> | Se incluyen librerías |
| #include<stdlib.h> |  |
| int sumar(int\* a,int\* b); | Se declara la variable sumar y sus parámetros |
| main (){ |  |
| int\* | Se le asigna a num1 4 bytes de memoria dinámica. |
| num1=(int\*)malloc(sizeof(int)); |  |
| int\* | Se le asigna a num2 4 bytes de memoria dinámica. |
| num2=(int\*)malloc(sizeof(int)); |  |
| \*num1=5; | Al apuntador num1 se le da el valor de 5. |
| \*num2=8; | Al apuntador num2 se le da el valor de 8. |
| int total=sumar(num1,num2); | Se llama a la función suma con los valores de 5 y 8. |
| } |  |
| int sumar(int\* a, int\* b){ |  |
| int c=0; |  |
| c=(\*a)+(\*b); | En la función suma se suma el 5 y 8, que es el valor de los apuntadores. |
| return c; | Regresa c. |
| } |  |

**Ejercicio 3:**

3. Programe los códigos del punto 1 y 2 e imprima en ambos casos el total, num1 y num2. Posteriormente responda las siguientes preguntas:

1. En el programa de paso por valor, ¿Qué sucede si modifica el valor de la variable a en la función sumar con respecto al valor de num1?

Se modifica el valor de la suma.

1. En el programa de paso por valor, ¿Qué sucede si modifica el valor de la variable total después del llamado de función con respecto al valor de c en la función sumar?

El valor que se devuelve es diferente.

1. En el programa de paso por referencia, ¿Qué sucede si al inicio de la función sumar introduce la línea de código a=4;?

El valor de a se cambia a 4.

1. En el programa de paso por referencia, ¿Qué sucede si al inicio de la función sumar introduce la línea de código \*a=4;?

Se guarda en la dirección \*a el valor de 4.

1. Calcule el uso de memoria para ambos programas y exponga la diferencia.

Se ocupa más espacio de memoria en el segundo programa por la extensión.