**Práctica 6**

1. Describa línea por línea el código de paso por valor siguiente:

|  |  |
| --- | --- |
| **Código paso por valor** | **Descripción** |
| #include <stdio.h> | Librería estándar para algunas funciones |
| int sumar(int a,int b); | Se declara la función con nombre suma |
| main (){ | Inicio del main o función principal |
| int num1; | Declaración variable entera uno |
| int num2; | Declaración variable entera dos |
| num1=5; | Asignación de valor al entero 1 |
| num2=8; | Asignación de valor al entero 2 |
| int total=sumar(num1,num2); | Declaración del entero total y llamado a la función suma con los valores de entero 1 y entero 2 |
| } | Fin de la función principal |
| int sumar(int a, int b){ | Inicio de la función sumar |
| int c=0; | Declaración entero auxiliar |
| c=(a)+(b); | Realización de suma con operadores aritméticos |
| return c; | Regreso del valor de la suma a donde se llamo la función |
| } | Fin de la función sumar |

2. Describa línea por línea el código de paso por referencia siguiente:

|  |  |
| --- | --- |
| **Código paso por referencia** | **Descripción** |
| #include <stdio.h> | Agrega la entrada y salida estándar |
| #include<stdlib.h> | Agrega la librería estándar |
|  |  |
| int sumar(int\* a,int\* b); | Declaración del prototipo de la función suma que recibe dos apuntadores a entero |
|  |  |
| main (){ | Inicio de la función principal |
| int\*  num1=(int\*)malloc(sizeof(int)); | Declaración de apuntador 1 a entero que es igual a un bloque de la memoria con tamaño de un entero (4 Bytes) |
| int\*  num2=(int\*)malloc(sizeof(int)); | Declaración de apuntador 2 a entero que es igual a un bloque de la memoria con tamaño de un entero (4 Bytes) |
| \*num1=5; | Asignación de valor al apuntador 1 |
| \*num2=8; | Asignación de valor al apuntador 2 |
| int total=sumar(num1,num2); | Declaración de entero total y llamado a la función sumar |
| } | Fin de función principal |
|  |  |
| int sumar(int\* a, int\* b){ | Inicio de la función suma |
| int c=0; | Declaración de entero auxiliar para realizar la suma |
| c=(\*a)+(\*b); | Realización de suma con operadores aritméticos mediante el uso de los valores contenidos en la memoria que esta asignada para a y b |
| return c; | Regresa en valor de la suma a donde fue llamada la función |
| } | Fin de la función sumar |

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?

El valor se mantiene , solo “a” cambia asi como la suma. Pero el valor de “a” no .

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?

Ambos son independientes, entonces no le afecta en lo absoluto hasta que se vuelva a realizar el llamado de la función, al imprimirlos nos damos cuenta que lo que le hagamos a total no le afecta a c.

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

Manda un warning y comienza a haber errores.

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 modifica la parte de la memoria en donde esta contenida y por lo tanto se modifica en num1.

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

Es mayor para el caso del programa que pasa por valor.