**Actividad 6**

**CARRERA:** Ingeniería en Computación

**NOMBRE:** *Efrain Robles Pulido*

**CÓDIGO:** 221350095

**MATERIA:** Traductores de Lenguajes I

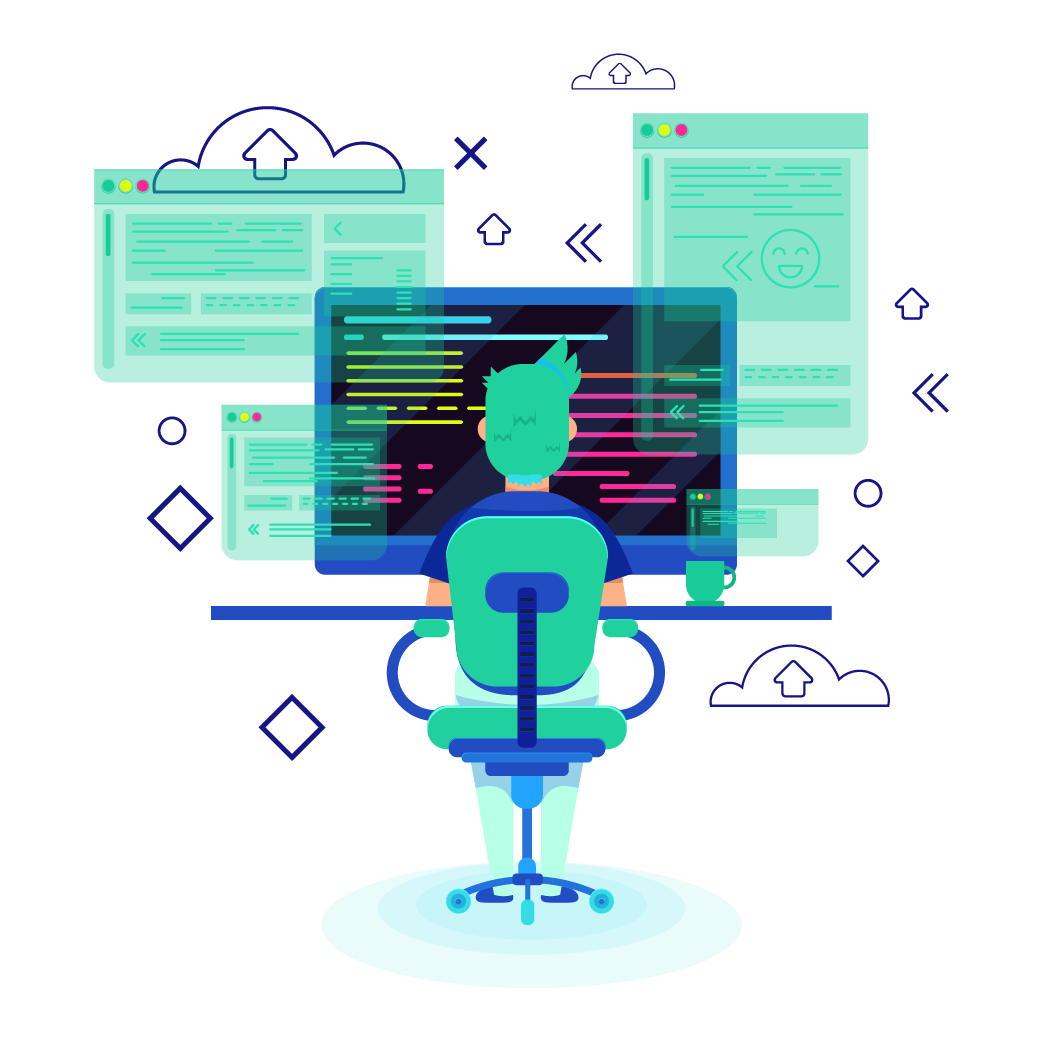
**MAESTRA:** José Juan Meza Espinoza

**SECCIÓN:** D09 **CALENDARIO:** 2023A

**UNIVERSIDAD DE GUADALAJARA**



**e**



**Desarrollo**

* Tomar el código de C y ASM que te compartió el profesor
* Agregar una función de ASM en ese código que contenga 3 entradas y 1 salida y que no sea ninguna de las funciones que ya están implementadas
* Poner el código en un archivo y el pantallazo de que si funciono tu código

Una vez obtenido el código C y ASM que compartió el profesor, se procedió a analizarlo y agregar una función aparte al código, en donde primero se nos pedirá 3 valores que serán necesarios para el calculo del volumen de un prisma rectangular, longitud, ancho y altura, para llevarlos a una función que realizara la operación de multiplicación entre ellos, como si fuera en ensamblador (ASM) por lo que ejecutamos las misma instrucciones, pero con diferente sintaxis, declarando un valor de resultado como entero, después en ASM mover el primer valor de entrada al registro Ax luego el segundo valor de entrada al registro Bx y utilizamos la instrucción mull para multiplicar el Bx con Ax automáticamente, después movemos el tercer valor de entrada a Bx y lo volvemos a multiplicar Bx con Ax, este resultado en Ax lo pasaremos a valor de salida para que nos regrese ese valor ya en código C. Para finalizar con la impresión del resultado de la operación.

**Código**:

#include <iostream>

**using namespace std**;

**int** suma( **int** $a, **int** $b ) {

**int** $result;

\_\_asm\_\_ \_\_volatile\_\_(

"movl %1, %%eax;"

"movl %2, %%ebx;"

"addl %%ebx,%%eax;"

"movl %%eax, %0;" : "=g" ( $result ) : "g" ( $a ), "g" ( $b ));

**return** $result ;

}

**int** resta( **int** $a, **int** $b) {

**int** $result;

\_\_asm\_\_ \_\_volatile\_\_(

"movl %1, %%eax;"

"movl %2, %%ebx;"

"subl %%ebx,%%eax;"

"movl %%eax, %0;" : "=g" ( $result ) : "g" ( $a ), "g" ( $b ));

**return** $result ;

}

**int** volumen(**int** $lon, **int** $anc, **int** $alt) {

**int** $result;

\_\_asm\_\_ \_\_volatile\_\_(

"movl %1, %%eax;"

"movl %2, %%ebx;"

"mull %%ebx;"

"movl %3, %%ebx;"

"mull %%ebx;"

"movl %%eax, %0;"

: "=g" ( $result ) : "g" ( $lon ), "g" ( $anc ), "g" ( $alt ));

**return** $result ;

}

**int** gcd( **int** $a, **int** $b ) {

**int** $result;

\_\_asm\_\_ \_\_volatile\_\_(

"movl %1, %%eax;"

"movl %2, %%ebx;"

"CONTD: cmpl $0, %%ebx;"

"je DONE;"

"xorl %%edx, %%edx;"

"idivl %%ebx;"

"movl %%ebx, %%eax;"

"movl %%edx, %%ebx;"

"jmp CONTD;"

"DONE: movl %%eax, %0;" : "=g" ( $result ) : "g" ( $a ), "g" ( $b ));

**return** $result ;

}

**int** main(**int** argc, **char**\*\* argv) {

**int** a,b,c;

**cout**<<"\nDigite el numero a: ";

**cin**>>a;

**cout**<<"\nDigite el numero b: ";

**cin**>>b;

c=suma(a,b);

**cout**<<"\nEl resultado de la suma de "<<a<<"+"<<b<<"="<<c<<"\n";

c=resta(a,b);

**cout**<<"\nEl resultado de la resta de "<<a<<"-"<<b<<"="<<c<<"\n";

c=gcd(a,b);

**cout**<<"\nEl resultado GCD("<<a<<","<<b<<")="<<c<<"\n";

**asm**("subl %%ebx, %%eax;"

"movl %%eax, %%ecx;"

: "=c" (c)

: "a" (a), "b" (b)

: /\* lista clobber vacia \*/

);

printf("a = %d\nb = %d\nc = %d\n", a, b, c);

**asm**("addl %%ebx, %%eax;"

"movl %%eax, %%ecx;"

: "=c" (c)

: "a" (a), "b" (b)

: /\* lista clobber vacia \*/

);

printf("a = %d\nb = %d\nc = %d\n", a, b, c);

/\*La lista clobber es una lista de cadenas separadas por comas.

Cada cadena es el nombre de un registro que el cÃ³digo ensamblador

modifica potencialmente, pero para el cual el valor

final no es importante. \*/

**int** lon,anc,alt,res;

**cout**<<"\nDigite la longitud: ";

**cin**>>lon;

**cout**<<"\nDigite el ancho: ";

**cin**>>anc;

**cout**<<"\nDigite la altura: ";

**cin**>>alt;

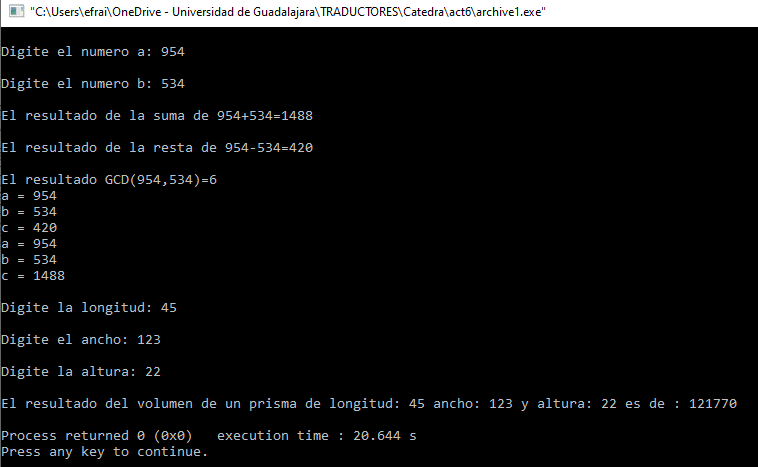
res=volumen(lon,anc,alt);

**cout**<<"\nEl resultado del volumen de un prisma de longitud: "<<lon

<<" ancho: "<<anc<<" y altura: "<<alt<<" es de : " <<res<<"\n";

**return** 0;

}



**Conclusión**

Me pareció muy confusa al principio de realizar ya que fue una mezcla de lenguaje extraño entre C y ASM, por lo que tuve que investigar por mi cuenta como funcionaba el lenguaje ASM en C. Por otro lado, no sabia que función realizar por lo que me fui a lo mas básico que podía hacer para practicar, el calculo de volumen de una figura. Para finalizar, fue muy educativo para mi ya que pude ver la verdadera aplicación que tiene el lenguaje ASM ya que nos permite configurar o codificar aun mas detallado que los lenguajes básicos como C o C++.

**Bibliografía:**

Brey, B. B. (2006). Microprocesadores Intel - 7 Edición (7a). Pearson Publications Company.