

Universidad de Carabobo
Facultad Experimental de Ciencias y Tecnología
Departamento de Computación
Arquitectura del Computador

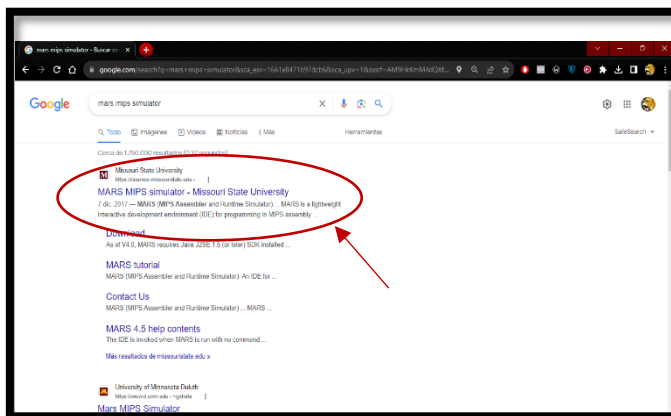
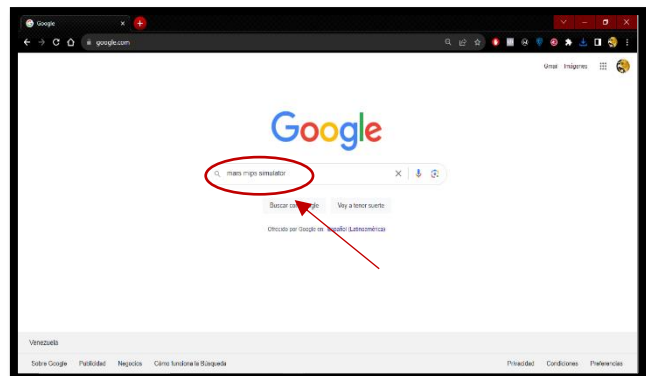
PRACTICA DE LABORATORIO 1

Realizado por:
Leonardo Moreno - C.I.:28330952
Luis Romero – C.I.:26729561

GUIA DE INSTALACION

PASO #1

La primera parte de la instalación en caso de hacerlo con Windows es dirigirse al buscador de tu preferencia en este caso Google Chrome y colocar "MARS MIPS SIMULATOR" y le damos a buscar.

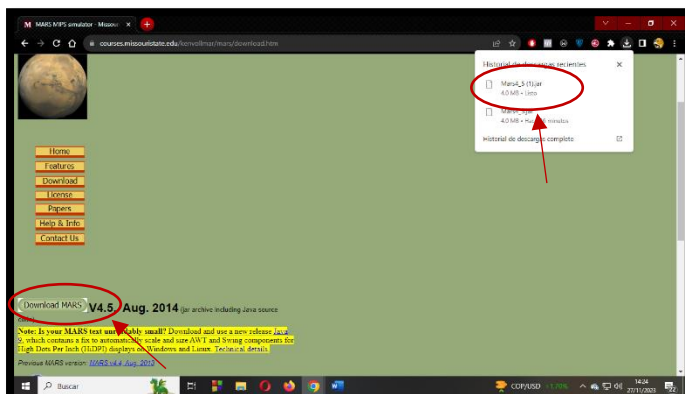
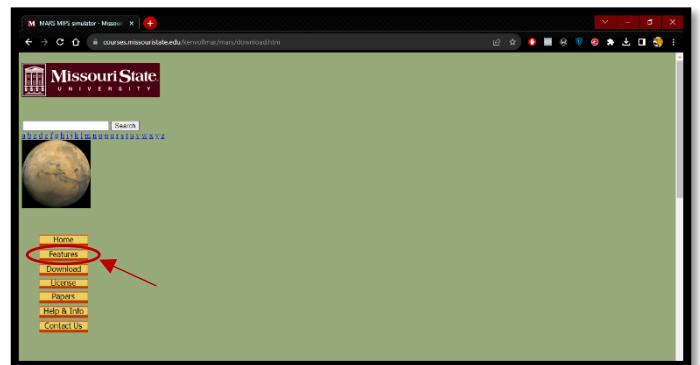


PASO #2

En este paso buscaremos el link de la universidad "Missouri state university" que por lo general esta en la primera posición y entramos a la pagina web.

PASO 3

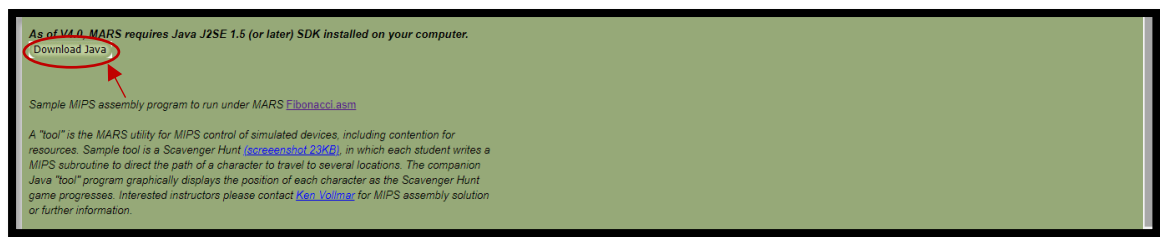
Entramos en la página y localizamos el botón que dice descargar o en este caso "download" para que nos mande a la parte donde podremos iniciar la descarga.



PASO #4

Le daremos click a la parte que dice "Download MARS V4.5" luego de esto el programa se empezara a descargar automáticamente en la parte de descargas de la pc.

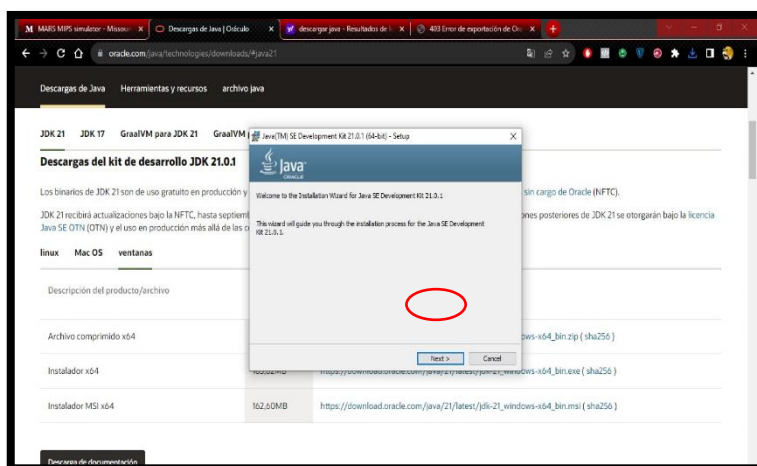
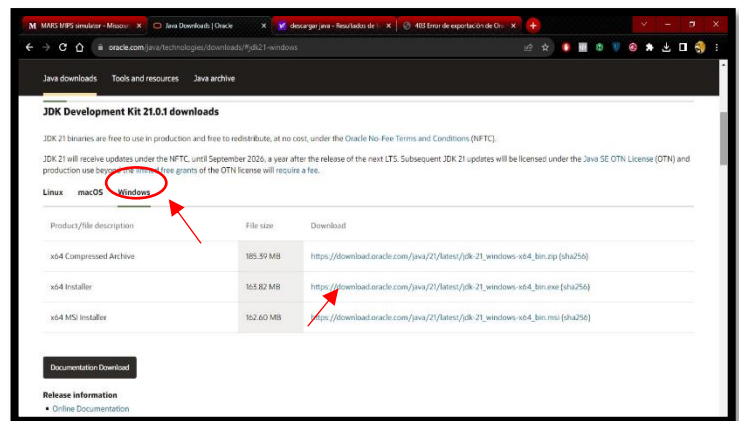
PASO #5



Vamos mas abajo en la pagina y le damos a donde dice "download java" para descargar el complemento necesario para poder instalar el programa para que no nos de ningún error el programa.

PASO #6

Una vez ingresado a la página buscaremos la parte de "download java" y nos dirigiremos a la parte de Windows y le daremos click al enlace que se descargara automáticamente, una vez hecho esto ejecutamos el archivo como administrador y procedemos con la instalación.

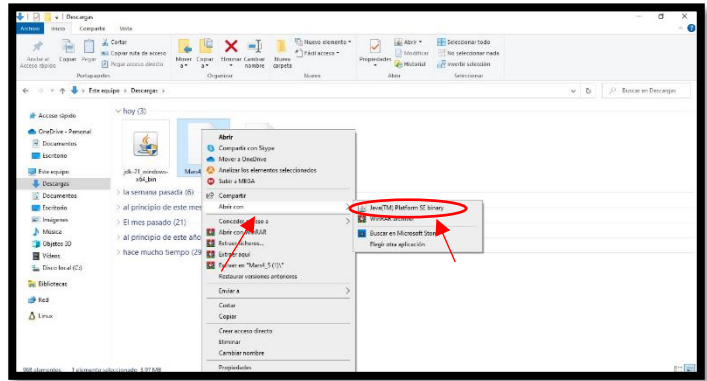


PASO #7

Iniciamos con el proceso de instalación, aceptamos los términos y seleccionamos la ubicación donde queremos que se instale el complemento de java para que pueda correr el programa

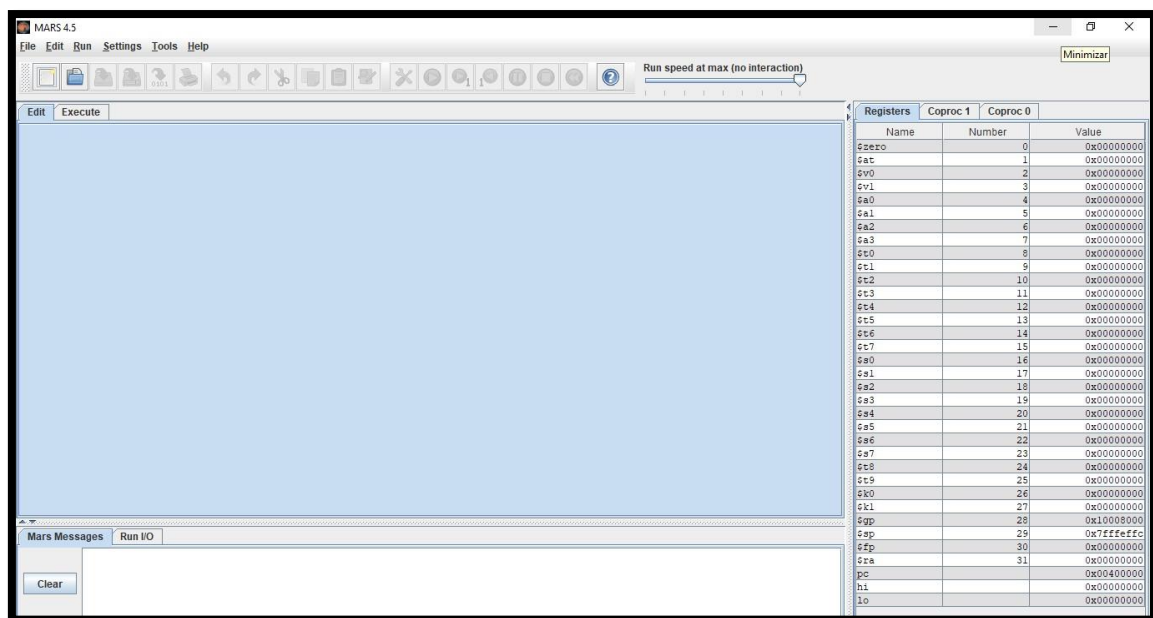
PASO #8

Damos click derecho sobre la aplicación, le damos a “abrir con” y abriremos el programa con el complemento que acabamos de instalar.

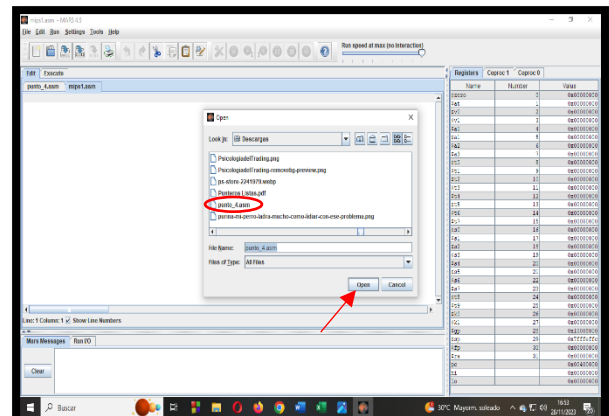
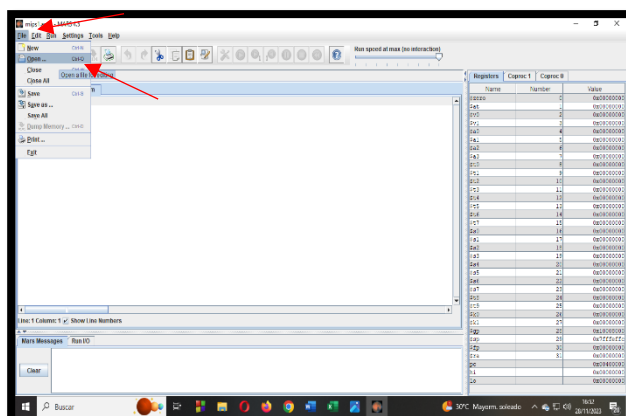


PASO #9

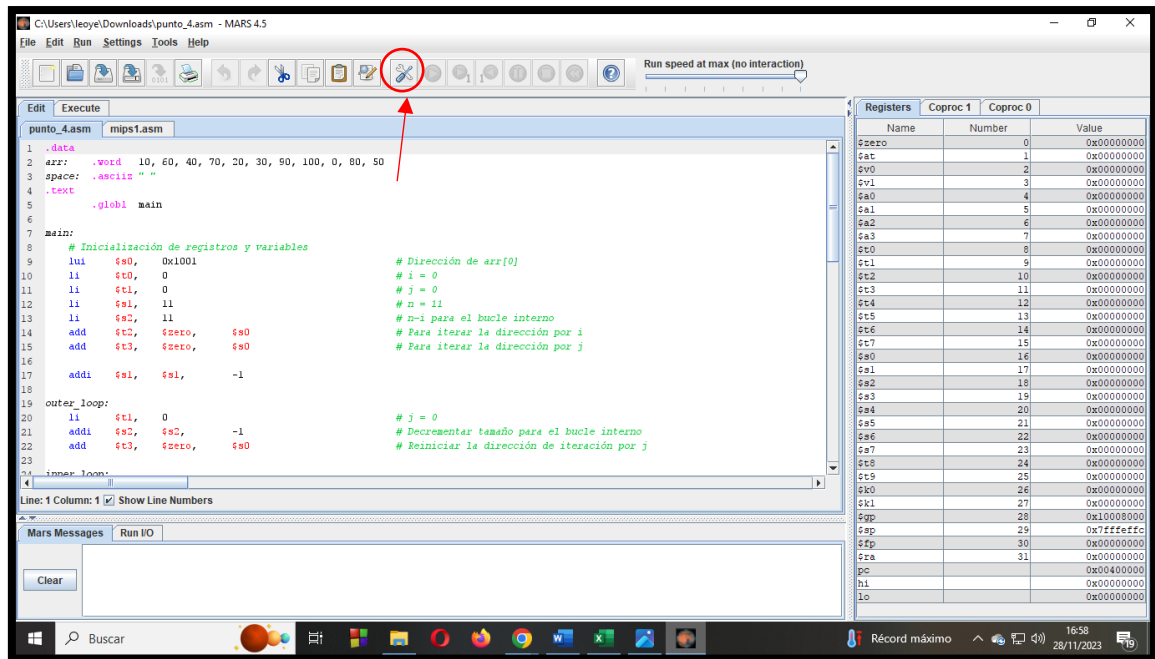
Esperamos a que inicie el programa para trabajar con él. En este caso haremos una corrida de un programa para ver como cargar y ejecutar el programa.



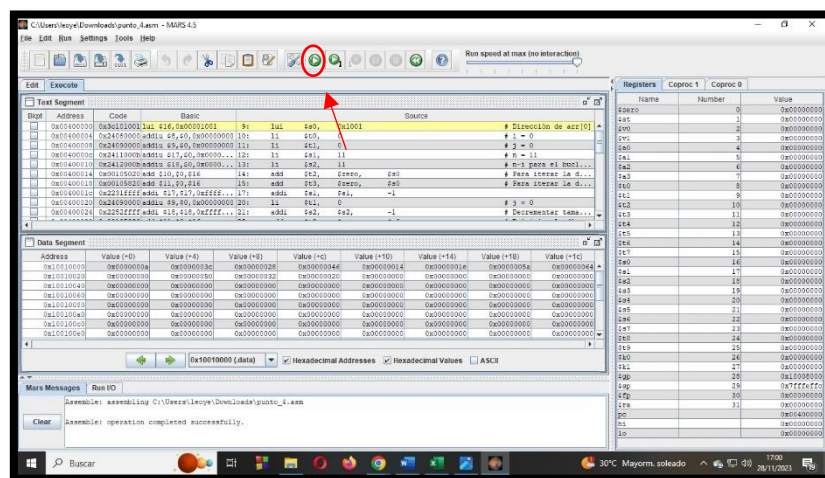
Le damos en “file” luego buscamos donde dice “open” y buscamos el archivo “.ams” que en este caso se llama “punto_4.ams” que es el algoritmo de ordenamiento burbuja (se explica mas detallado en el código como funciona)



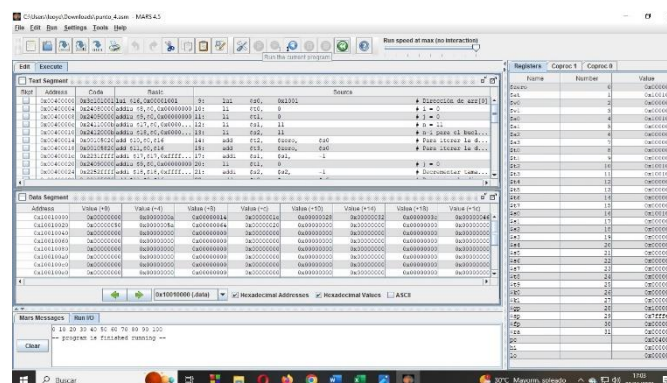
Así se vería un código ya abierto y para correrlo lo único que se tiene que hacer es darle al icono de “ASSAMBLE” para poder compilar y ver si tenemos errores antes de correrlo.



De la siguiente manera quedaría la compilación y lo único que queda es correrlo al icono “run” para que el programa se ejecute.



Para terminar con esta parte, este sería el resultado final de la corrida del código de ordenamiento burbuja



PUNTO #2

Código 1: Generar y Imprimir la Secuencia de Fibonacci

Objetivo

Este código tiene como objetivo calcular y almacenar los primeros 19 números de Fibonacci en un array y luego imprimirlos en una sola línea.

Características Clave

- **Entrada de Usuario Opcional:** Aunque la entrada de usuario está comentada, el código permite al usuario especificar la cantidad de números de Fibonacci a generar.
- **Bucle de Cálculo de Fibonacci:** Utiliza un bucle (loop) para calcular los números de Fibonacci y almacenarlos en un array.
- **Subrutina de Impresión:** La subrutina print está bien definida y se encarga de imprimir los elementos del array en una línea.

Estructura y Comprobaciones

- **Estructura Clara y Eficiente:** La estructura del código es clara y eficiente, utilizando un bucle para el cálculo y almacenamiento de los números de Fibonacci.
- **Comprobaciones de Límites:** Incluso si la entrada de usuario está comentada, el código incluye comprobaciones de límites para garantizar que la cantidad de números de Fibonacci solicitada esté dentro de los límites permitidos.

Diferencias con el Código Académico

- Este código utiliza un enfoque iterativo para calcular los números de Fibonacci y almacenarlos en un array, en contraste con el enfoque recursivo del código académico.

Código 2: Traversal de Matriz en Orden de Filas

Objetivo

Este código realiza el traversal de una matriz de 16x16 en orden de filas, almacenando valores incrementales en cada elemento.

Características Clave

- **Estructura de Doble Bucle:** Utiliza un doble bucle anidado para recorrer la matriz en orden de filas.
- **Cálculo de Desplazamiento Efectivo:** El cálculo del desplazamiento (offset) para acceder a elementos de la matriz es eficiente y correcto, utilizando la multiplicación para garantizar el acceso adecuado.

Implementación

- **Declaración de Matriz y Variables:** Declara la matriz y las variables necesarias para el traversal, como el número de filas y columnas.
- **Bucle de Traversal:** Utiliza un bucle anidado para recorrer la matriz y almacenar valores incrementales.

Diferencias con el Código Académico

- A diferencia del código académico, este código sigue una estructura de doble bucle anidado para el traversal de la matriz en orden de filas, mientras que el código académico tiene errores lógicos en su enfoque.

Código 3: Traversal de Matriz en Orden de Columnas

Objetivo

Este código realiza el traversal de una matriz de 16x16 en orden de columnas, almacenando valores incrementales en cada elemento.

Características Clave

- **Estructura de Doble Bucle:** Al igual que el código anterior, utiliza un doble bucle anidado, pero en este caso, recorre la matriz en orden de columnas.
- **Cálculo de Desplazamiento Efectivo:** Al igual que el código anterior, el cálculo del desplazamiento (offset) para acceder a elementos de la matriz es eficiente y correcto.

Implementación

- **Declaración de Matriz y Variables:** Declara la matriz y las variables necesarias para el traversal, como el número de filas y columnas.
- **Bucle de Traversal:** Utiliza un bucle anidado para recorrer la matriz y almacenar valores incrementales.

Diferencias con el Código Académico

- **En comparación con el código académico, este código sigue una estructura de doble bucle anidado para el traversal de la matriz en orden de columnas, mientras que el código académico tiene errores lógicos en su enfoque.**