Asignatura	Datos del alumno	Fecha
Estructura de Computadores	Apellidos: Gómez Pérez	7 de septiembre de 2024
	Nombre: Bárbara Catalina	

LABORATORIO #1: SIMULACIÓN Y OPTIMIZACIÓN DE UN PROGRAMA EN UN PROCESADOR ESCALAR SEGMENTADO

FUNDACION UNIVERSITARIA INTERNACIONAL DE LA RIOJA

INGENIERÍA INFORMÁTICA

Bárbara Catalina Gómez Pérez

Profesor: Deivis Eduard Ramirez Martinez

BOGOTA, D.C.
2024

Asignatura	Datos del alumno	Fecha
Estructura de Computadores	Apellidos: Gómez Pérez	7 de septiembre de 2024
	Nombre: Bárbara Catalina	

ÍNDICES DE CONTENIDO

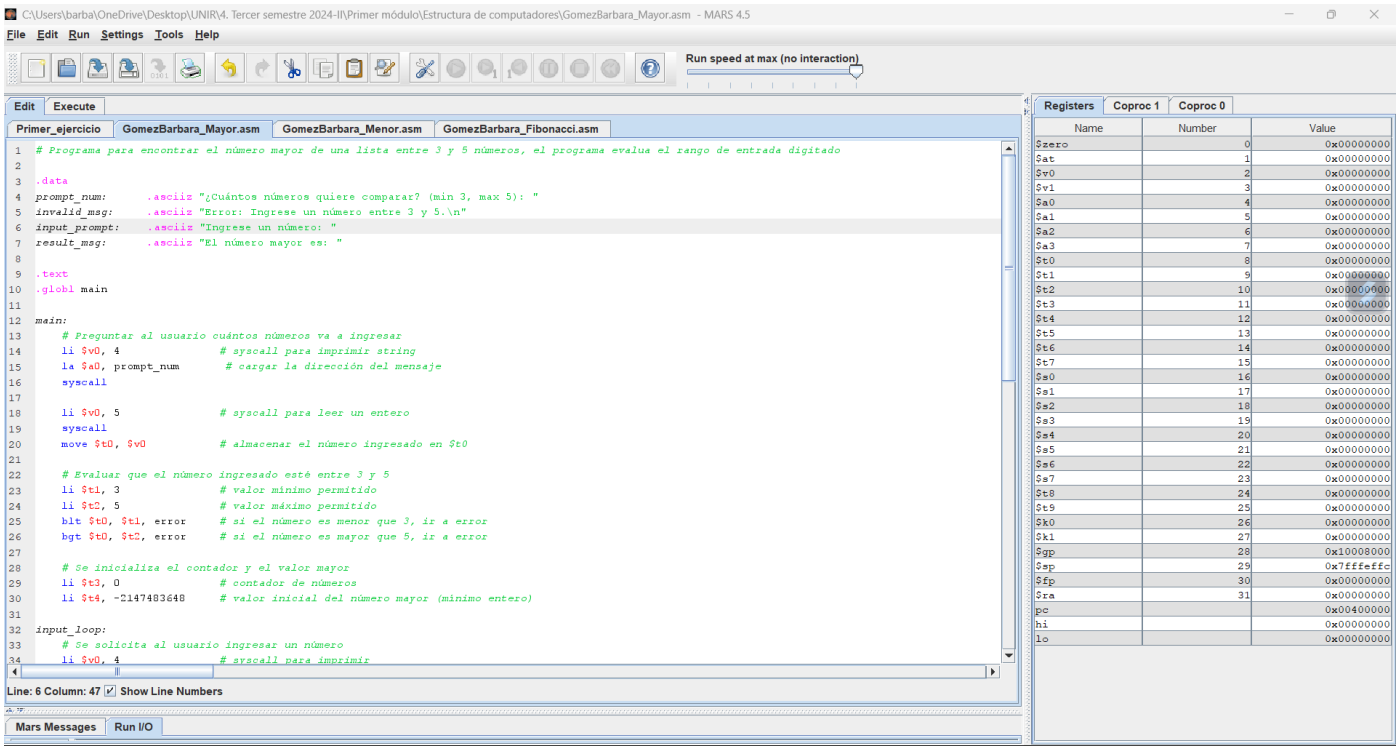
Asignatura	Datos del alumno	Fecha
Estructura de Computadores	Apellidos: Gómez Pérez	7 de septiembre de 2024
	Nombre: Bárbara Catalina	

1. Introducción

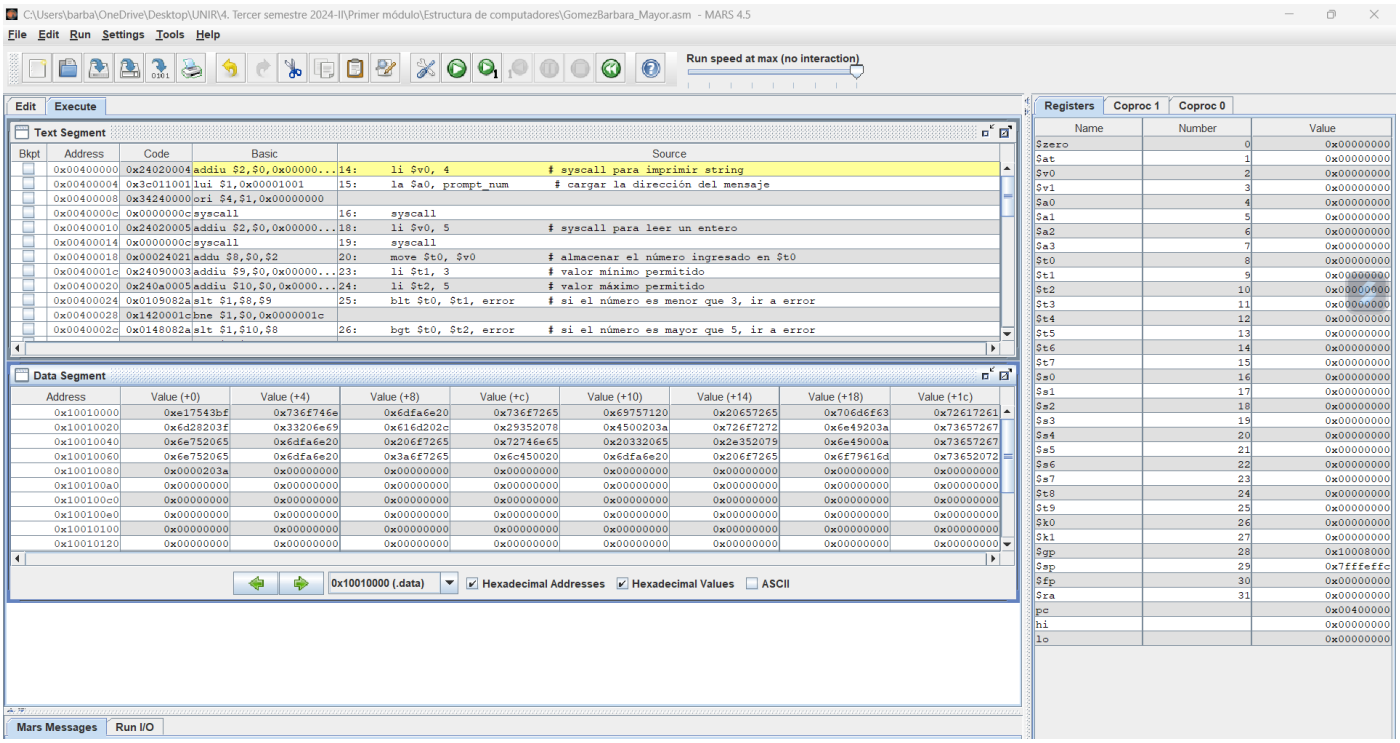
En este laboratorio se desarrollaron tres scripts en lenguaje ensamblador MIPS. El objetivo principal de la práctica de laboratorio es evaluar la capacidad de interacción con el usuario a través de la consola, ejecutar cálculos matemáticos básicos y utilizar estructuras de control simples. Cada script creado cumple una función específica como comparar números hasta crear secuencias numéricas, utilizando operaciones aritméticas y lógicas elementales, así como manejo de bucles y condiciones. Este laboratorio también nos brinda la oportunidad de entender cómo opera el procesador internamente y de qué manera se llevan a cabo las instrucciones con lenguaje de programación de bajo nivel.

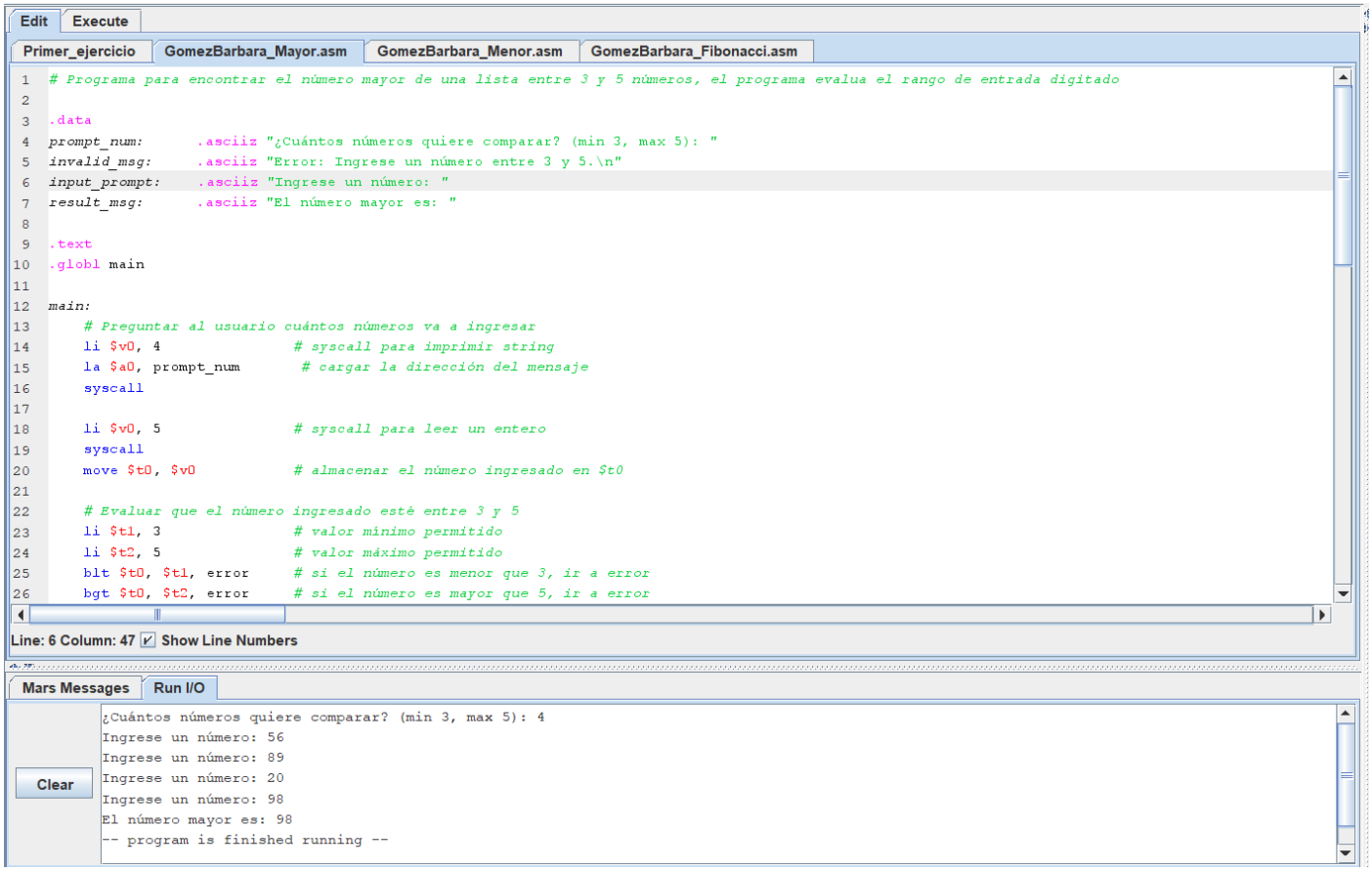
2. Desarrollo de la actividad.

2.1. Solución script #1: Número mayor (mínimo 3 números).



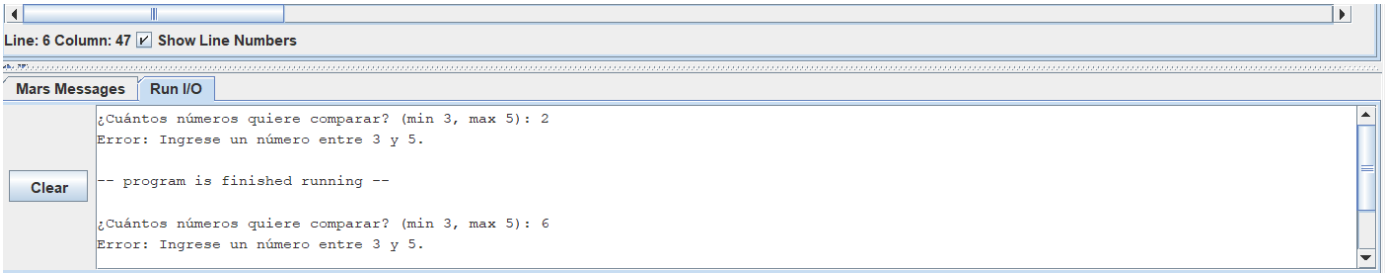
Asignatura	Datos del alumno	Fecha
Estructura de Computadores	Apellidos: Gómez Pérez	7 de septiembre de 2024
	Nombre: Bárbara Catalina	





Asignatura	Datos del alumno	Fecha
Estructura de Computadores	Apellidos: Gómez Pérez	7 de septiembre de 2024
	Nombre: Bárbara Catalina	

El primer script le solicita al usuario que digite entre 3 y 5 números, con el fin de encontrar el número mayor ingresado. El programa inicia preguntando al usuario que indique cuántos números quiere comparar. Si el número ingresado está fuera del rango permitido (menos de 3 o más de 5 números), el programa muestra un mensaje de error y termina (Peralta, 2014).



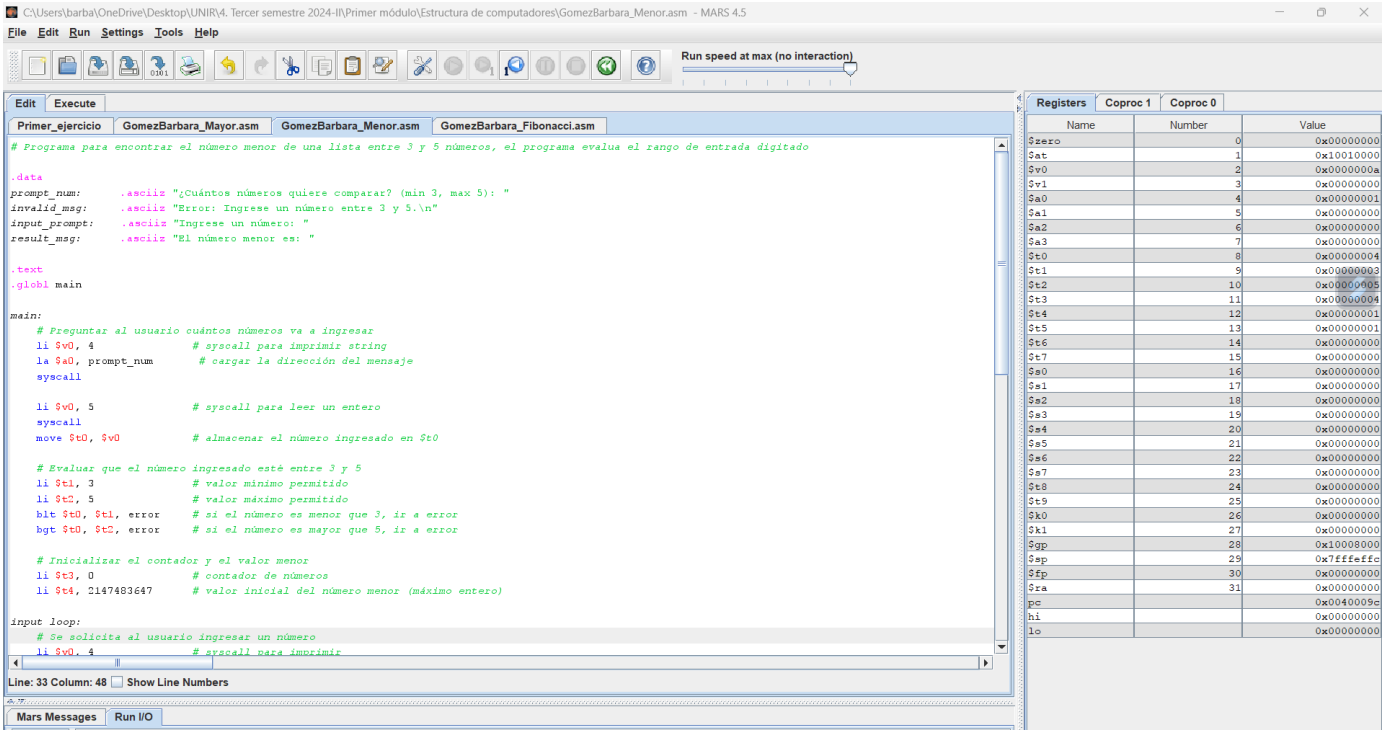
Ejemplo mensaje de error cuando se digita un número fuera del rango permitido.

Si la cantidad ingresada es válida, el usuario podrá ingresar los números uno por uno, de esta manera el programa comparará cada entrada con el número mayor almacenado. Una vez se han ingresado todos los números, el programa mostrará el mayor en la consola.

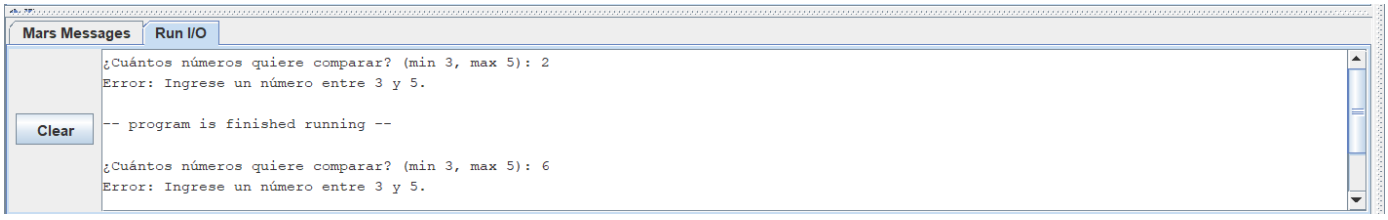
Enlace programa Git Hub:
https://github.com/bcgomez/Lab1_EstructuraComputadores/blob/main/GomezBarbara_Mayor.asm

Asignatura	Datos del alumno	Fecha
Estructura de Computadores	Apellidos: Gómez Pérez	7 de septiembre de 2024
	Nombre: Bárbara Catalina	

2.2. Solución script #2: Número menor (mínimo 3 números y máximo 5 números).

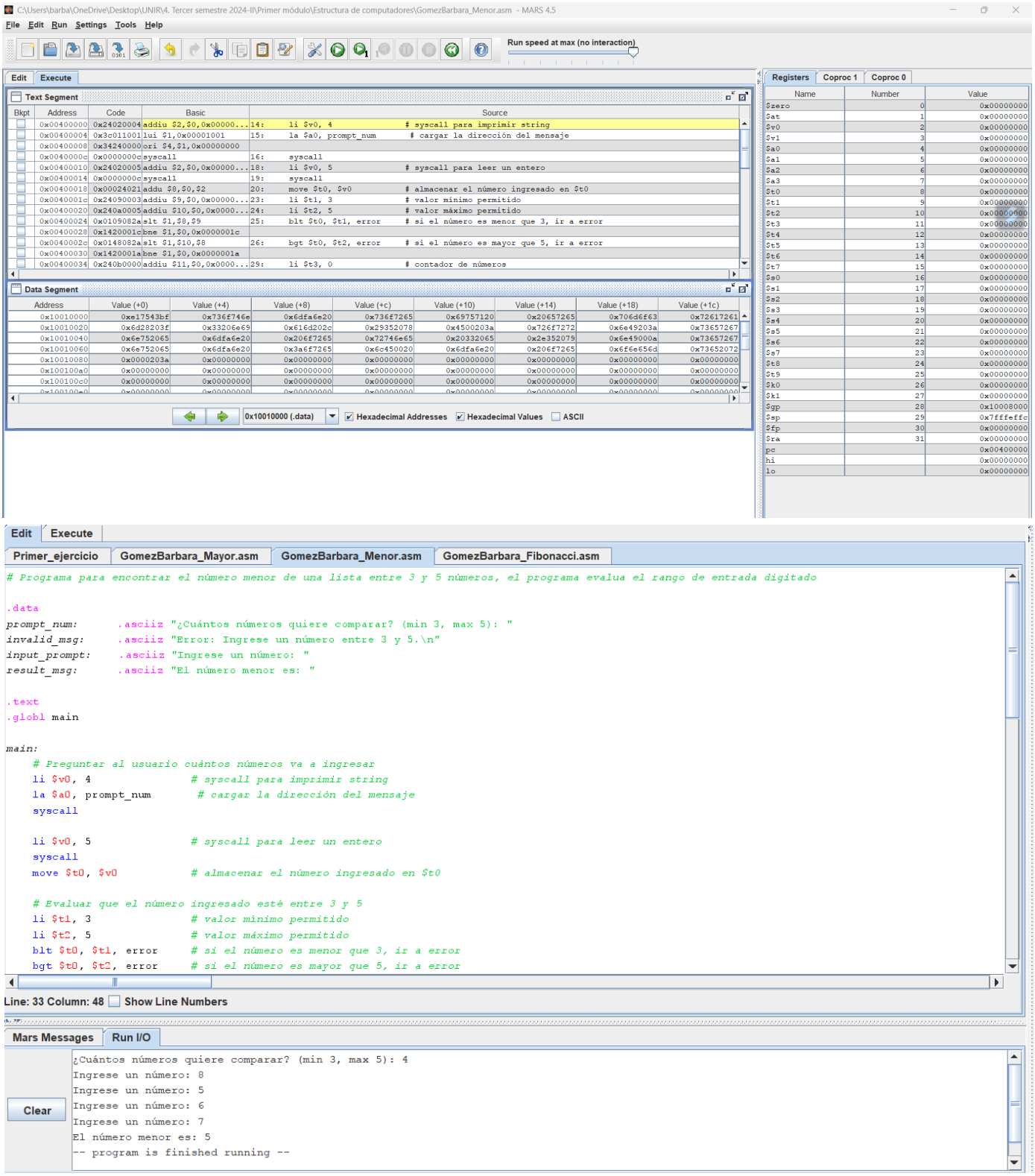


El segundo script le solicita al usuario que ingrese entre 3 y 5 números, con el propósito de encontrar el número menor. El desarrollo del programa es muy parecido al anterior script, porque primero se evalúa que el número de entradas esté dentro del rango (entre 3 y 5 números), luego, se le solicita al usuario que ingrese los números uno por uno. A medida que se ingresan los números, cada número se compara con el menor almacenado. Cuando se terminan de ingresar todos los números, el programa imprime el menor.



Ejemplo mensaje de error cuando se digita un número fuera del rango permitido.

Asignatura	Datos del alumno	Fecha
Estructura de Computadores	Apellidos: Gómez Pérez	7 de septiembre de 2024
	Nombre: Bárbara Catalina	

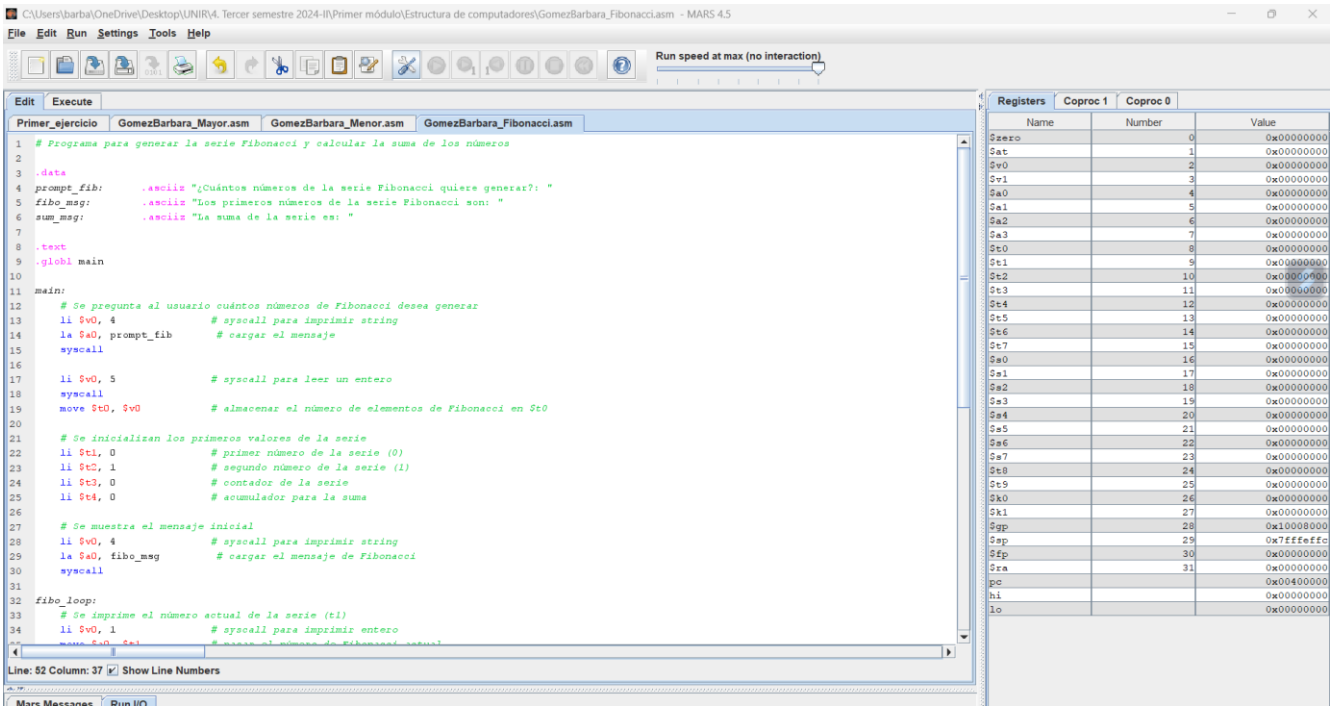


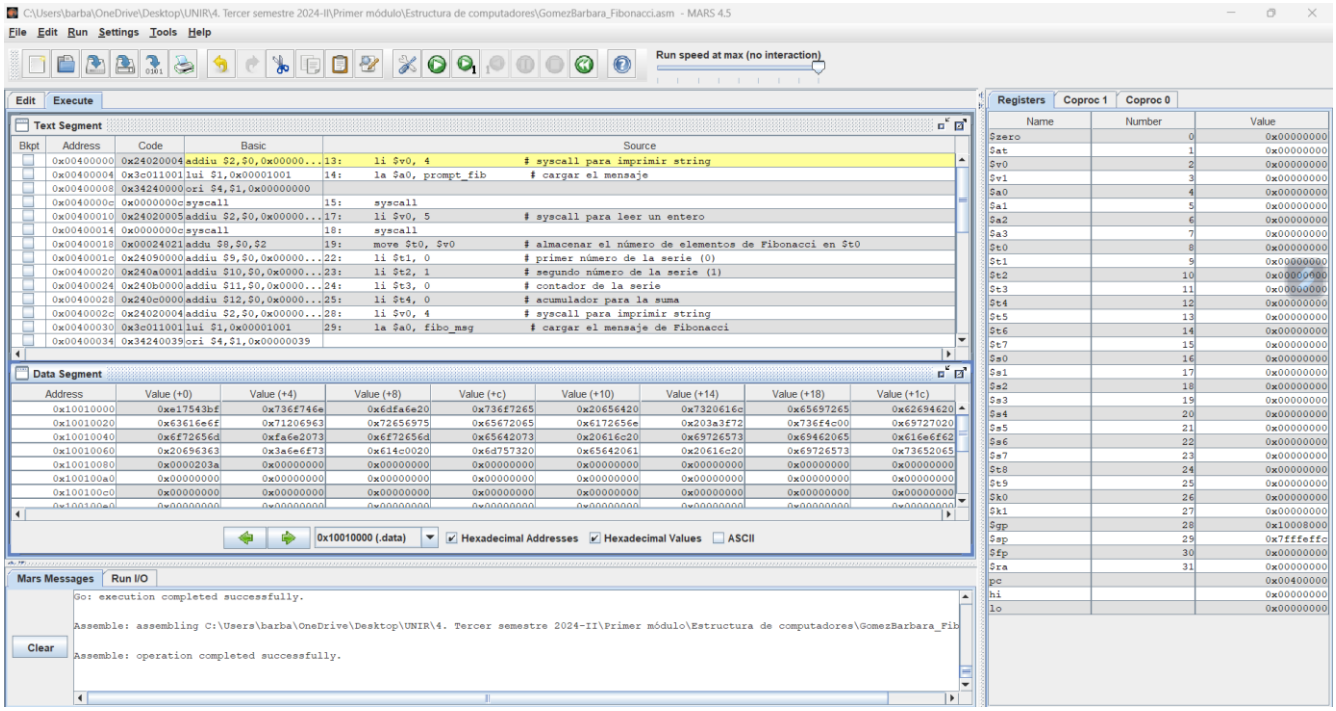
Enlace Git Hub:

https://github.com/bcgomez/Lab1_EstructuraComputadores/blob/main/GomezBarbara_Menor.asm

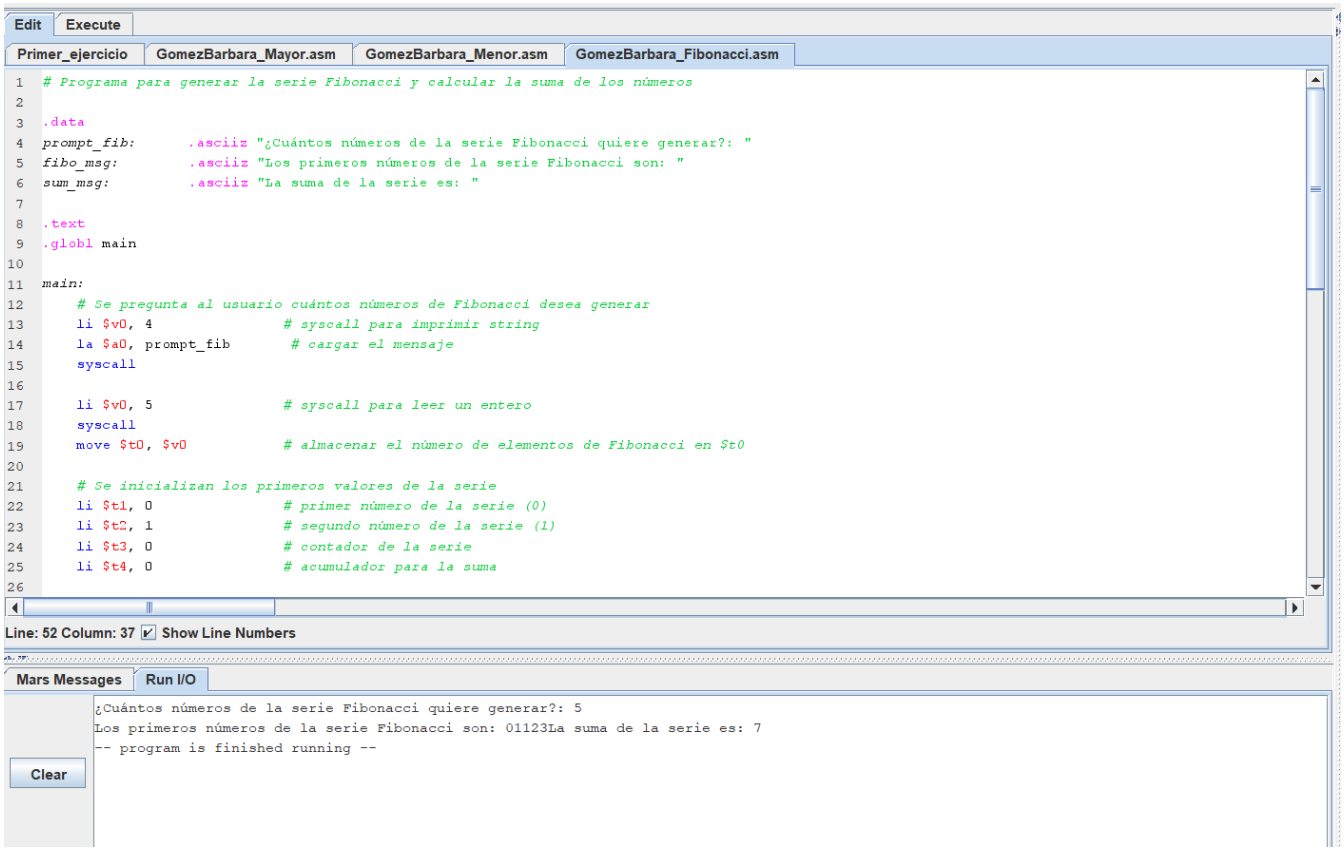
Asignatura	Datos del alumno	Fecha
Estructura de Computadores	Apellidos: Gómez Pérez	7 de septiembre de 2024
	Nombre: Bárbara Catalina	

2.3 Solución script #3: Serie Fibonacci.





Asignatura	Datos del alumno	Fecha
Estructura de Computadores	Apellidos: Gómez Pérez	7 de septiembre de 2024
	Nombre: Bárbara Catalina	



Enlace programa GitHub:
https://github.com/bcgomez/Lab1_EstructuraComputadores/blob/main/GomezBarbara_Fibonacci.asm

En este script, el programa genera la serie Fibonacci hasta un número de términos declarado por el usuario. Así como en los anteriores scripts, se evalúa que el número ingresado por el usuario esté dentro del rango (al menos 1 término, para esta serie). Una vez digitados los números de la serie, se genera los números de la serie Fibonacci a través de una suma sencilla de los dos términos anteriores y los va mostrando en la consola a medida que se calculan.

Fórmula empleada para calcular la serie de Fibonacci:

$$f_n = f_{n-1} + f_{n-2}$$

(Escobar, 2020)

Al final, el programa también calcula la suma de todos los números de la serie y muestra el resultado.

3. Conclusiones.

- En el desarrollo del laboratorio logré comprender las bases del lenguaje ensamblador MIPS, lo cual me permitió interactuar con el hardware a través de las instrucciones y comandos de este lenguaje de bajo nivel. De esta forma, el programa MARS me facilitó observar cómo un procesador ejecuta acciones sencillas como comparaciones y cálculos matemáticos entre números.
- En todos los scripts creados se pusieron en práctica validaciones para verificar que se cumplieran los límites de los rangos en la entrada de datos dada por los usuarios. Por medio de estas evaluaciones, se aseguró que en los dos primeros programas no de aceptara valores fuera del rango, imprimiendo el mensaje de error: “Error: ingrese un número entre 3 y 5”.
- La interacción de la consola con el usuario se fortalece a través del uso de syscalls para la entrada y salida de datos (Vollmar, 2006). Es importante resaltar el uso de este comando, ya que es decisivo para la creación de programas interactivos y entendibles para el usuario.
- Se demostró eficiencia en la ejecución de operaciones secuenciales, como en la implementación de bucles en la serie Fibonacci para acciones reiterativas. Asimismo, se utilizaron estructuras de control simples para evaluar la entrada de números y su comparación, mejorando así el rendimiento del programa.

4. Referencias

Escobar, E. V. (17 de Octubre de 2020). *Diagrama de flujo: Serie Fibonacci 1, 1, 2, 3, 5, 8, 13, 21,.* Obtenido de https://www.youtube.com/watch?app=desktop&v=xRna7mcBNLw

Peralta, A. (26 de Diciembre de 2014). *MIPS Tutorial 24 Checking If a Number is Less than Another slt.* Obtenido de https://www.youtube.com/watch?v=WF8jzQY0bh0&list=PL5b07qlmA3P6zUdDf-o97ddfpvPFuNa5A&index=24

Vollmar, K. (1 de Marzo de 2006). *MARS: An Education-Oriented MIPS Assembly Language Simulator.* Obtenido de file:///C:/Users/barba/Downloads/MarsAnEducation-Oriented_MIPS_AssemblyLanguaje%20(1).pdf