

1. Características principales de una máquina multinivel

- Una máquina multinivel se organiza en diferentes capas o niveles de abstracción.
- Cada nivel ofrece funciones específicas que facilitan la comunicación entre el usuario y el hardware.
- Permite simplificar el diseño y el uso de la computadora, ya que el usuario no interactúa directamente con el hardware.
- Cada nivel oculta la complejidad del nivel inferior y proporciona servicios más fáciles de utilizar.

2. Comunicación entre cada nivel de una máquina multinivel

- **Hardware:** Nivel más bajo, incluye los circuitos, memoria y procesadores.
- **Microarquitectura:** Controla cómo el hardware ejecuta instrucciones básicas, maneja registros y memoria caché.
- **Sistema Operativo:** Administra recursos de hardware y proporciona servicios al software. Se comunica con la microarquitectura mediante controladores y llamadas al sistema (syscalls).
- **Lenguaje de Alto Nivel:** Lenguajes como C, Java o Python que permiten al programador escribir instrucciones comprensibles. Estos se traducen a lenguaje máquina mediante compiladores o intérpretes.
- **Aplicación:** Programas que usa el usuario (ejemplo: navegadores, editores de texto). Se comunican con el sistema operativo a través de APIs, bibliotecas y syscalls.

Interfaces, traductores y protocolos:

- **Compiladores/Intérpretes:** Traducen de lenguaje de alto nivel a instrucciones máquina.
- **APIs y librerías:** Sirven de interfaz entre aplicaciones y sistema operativo.
- **Syscalls:** Protocolos de comunicación entre programas y el sistema operativo.

3. Características del lenguaje de bajo nivel

- Cercano al hardware y al lenguaje máquina.
- Alta velocidad de ejecución.
- Difícil de aprender y usar.
- Poco portable entre diferentes arquitecturas.
- Ejemplo: lenguaje ensamblador.

4. Características del lenguaje de alto nivel

- Fácil de comprender para los humanos.
- Portables entre diferentes sistemas y arquitecturas.
- Permiten escribir menos código para tareas complejas.
- Usan compiladores o intérpretes para traducirse al lenguaje máquina.
- Ejemplos: C, Python, Java.

5. Funcionamiento de las siguientes salidas del sistema (Syscalls) en arquitectura x86

- a. **Sys_exit (1):** Termina la ejecución de un programa y devuelve un código de salida al sistema operativo.
- b. **Sys_write (4):** Escribe datos en una salida, como la pantalla o un archivo. Necesita parámetros: descriptor de archivo, buffer y tamaño.
- c. **Sys_read (3):** Lee datos desde una entrada (como teclado o archivo) y los almacena en un buffer.

6. De acuerdo al código ensamblador anexo (también lo puedes encontrar aquí:

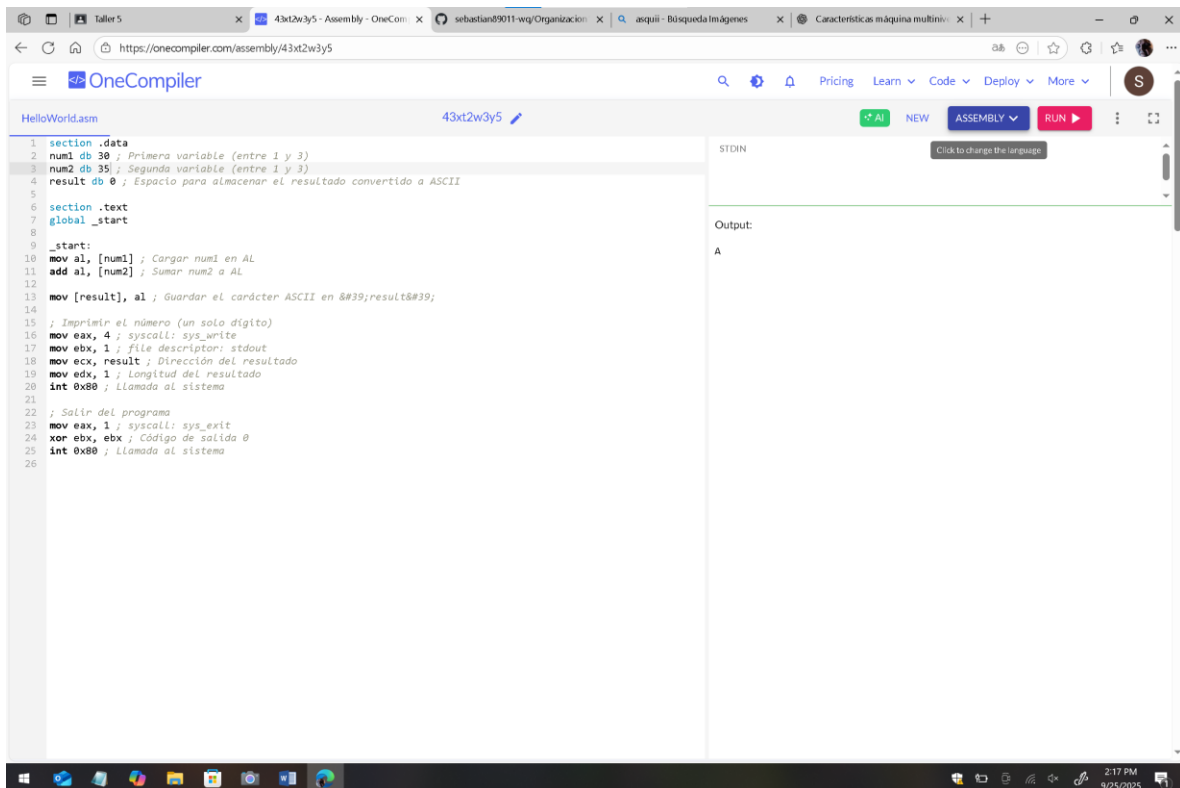
<https://onecompiler.com/assembly/43xsbvujq>), realiza

lo siguiente:

a. Modifica el código para que imprima los

siguientes caracteres utilizando solo sumas:

- A

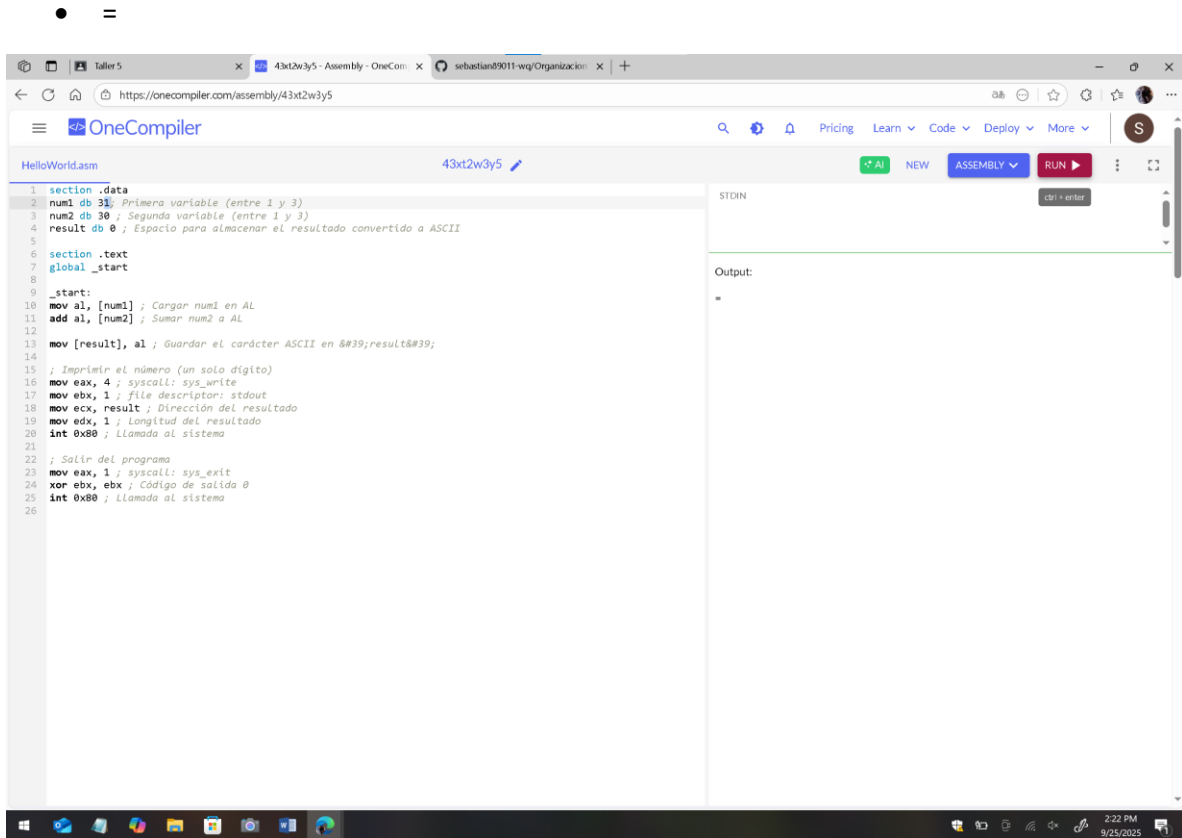
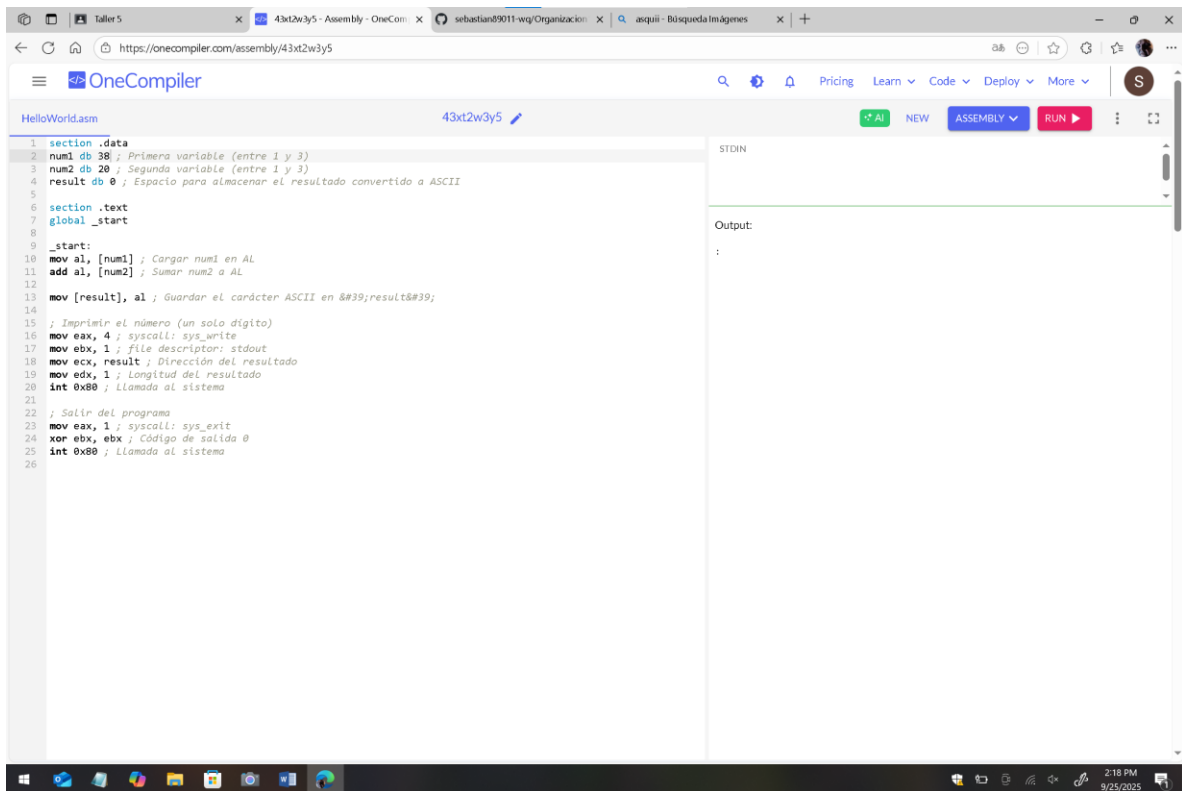


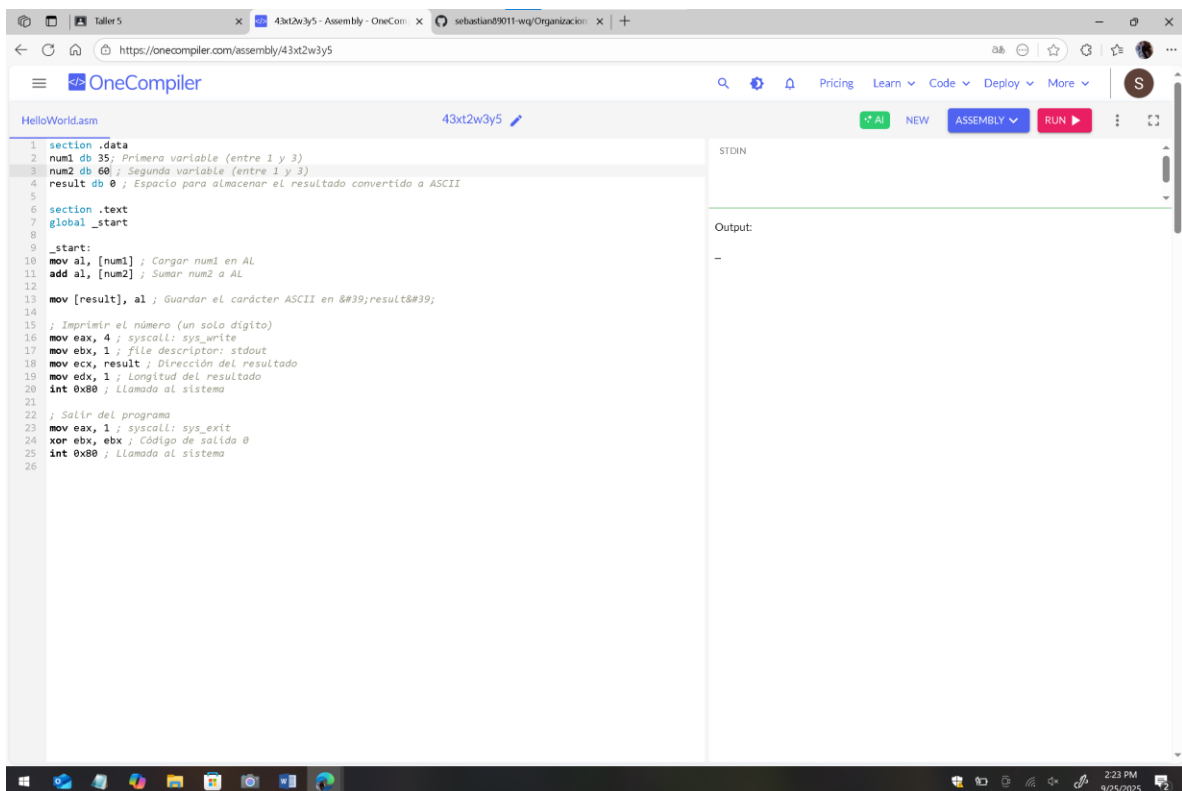
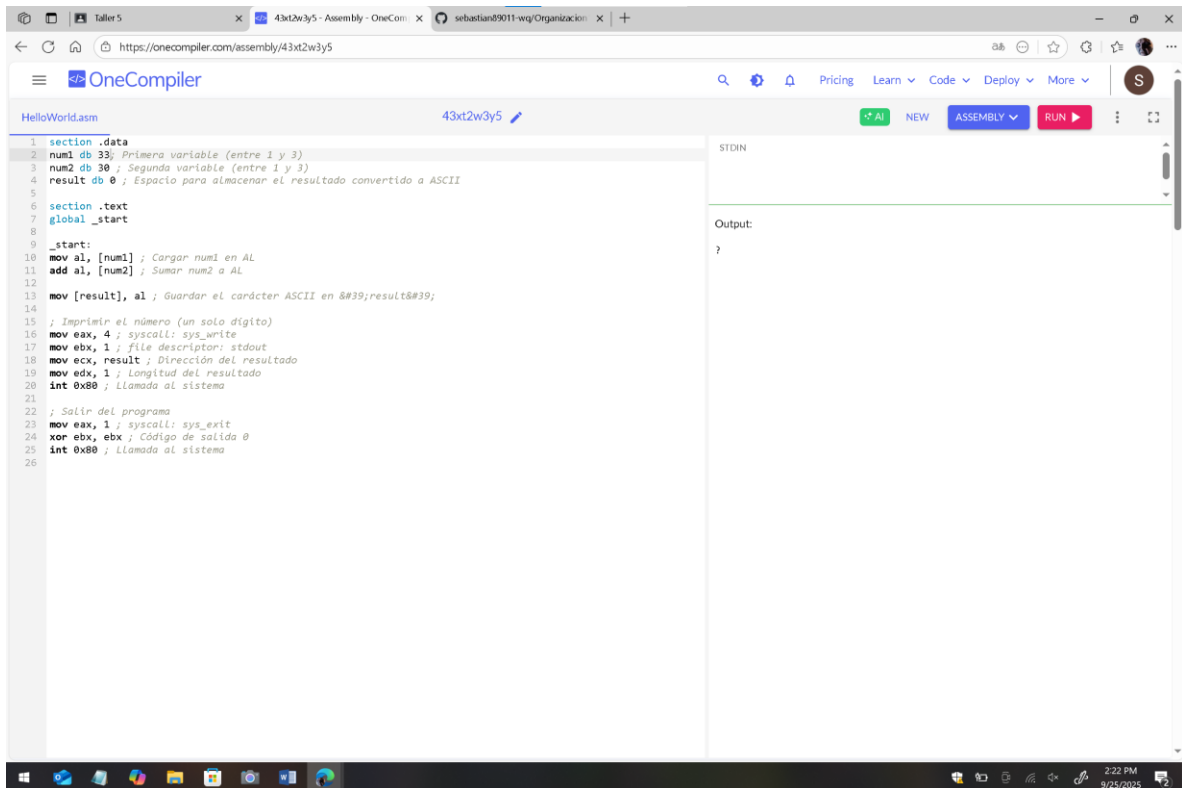
```
1 section .data
2 num1 db 30 ; Primera variable (entre 1 y 3)
3 num2 db 35 ; Segunda variable (entre 1 y 3)
4 result db 0 ; Espacio para almacenar el resultado convertido a ASCII
5
6 section .text
7 global _start
8
9 _start:
10 mov al, [num1] ; Cargar num1 en AL
11 add al, [num2] ; Sumar num2 a AL
12
13 mov [result], al ; Guardar el carácter ASCII en @#39;result&#39;
14
15 ; Imprimir el número (un solo dígito)
16 mov eax, 4 ; syscall: sys_write
17 mov ebx, 1 ; file descriptor: stdout
18 mov ecx, result ; Dirección del resultado
19 mov edx, 1 ; Longitud del resultado
20 int 0x80 ; Llamada al sistema
21
22 ; Salir del programa
23 mov eax, 1 ; syscall: sys_exit
24 xor ebx, ebx ; Código de salida 0
25 int 0x80 ; Llamada al sistema
26
```

Output:

A

- :





b. Ahora modificarlo para imprima los siguientes caracteres utilizando al menos una resta dentro del código:

- B

The screenshot shows the OneCompiler IDE with the following assembly code in the editor:

```
1 section .data
2 num1 db 96; Primera variable (entre 1 y 3)
3 num2 db 30; Segunda variable (entre 1 y 3)
4 result db 0; Espacio para almacenar el resultado convertido a ASCII
5
6 section .text
7 global _start
8
9 _start:
10 mov al, [num1]; Cargar num1 en AL
11 sub al, [num2]; Sumar num2 a AL
12
13 mov [result], al; Guardar el carácter ASCII en &#39;result&#39;;
14
15 ; Imprimir el número (un solo dígito)
16 mov eax, 4; syscall: sys_write
17 mov ebx, 1; file descriptor: stdout
18 mov ecx, result; Dirección del resultado
19 mov edx, 1; Longitud del resultado
20 int 0x80; Llamada al sistema
21
22 ; Salir del programa
23 mov eax, 1; syscall: sys_exit
24 xor ebx, ebx; Código de salida 0
25 int 0x80; Llamada al sistema
26
```

The output window on the right shows the character 'B' printed to the console.

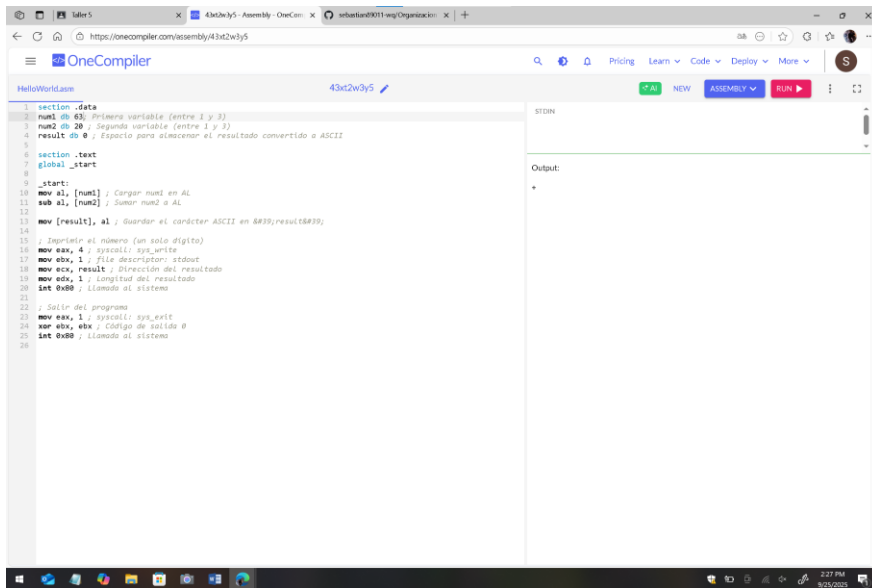
- X

The screenshot shows the OneCompiler IDE with the following assembly code in the editor:

```
1 section .data
2 num1 db 540; Primera variable (entre 1 y 3)
3 num2 db 30; Segunda variable (entre 1 y 3)
4 result db 0; Espacio para almacenar el resultado convertido a ASCII
5
6 section .text
7 global _start
8
9 _start:
10 mov al, [num1]; Cargar num1 en AL
11 sub al, [num2]; Sumar num2 a AL
12
13 mov [result], al; Guardar el carácter ASCII en &#39;result&#39;;
14
15 ; Imprimir el número (un solo dígito)
16 mov eax, 4; syscall: sys_write
17 mov ebx, 1; file descriptor: stdout
18 mov ecx, result; Dirección del resultado
19 mov edx, 1; Longitud del resultado
20 int 0x80; Llamada al sistema
21
22 ; Salir del programa
23 mov eax, 1; syscall: sys_exit
24 xor ebx, ebx; Código de salida 0
25 int 0x80; Llamada al sistema
26
```

The output window on the right shows the character 'X' printed to the console.

• +



OneCompiler

43x2w3y5

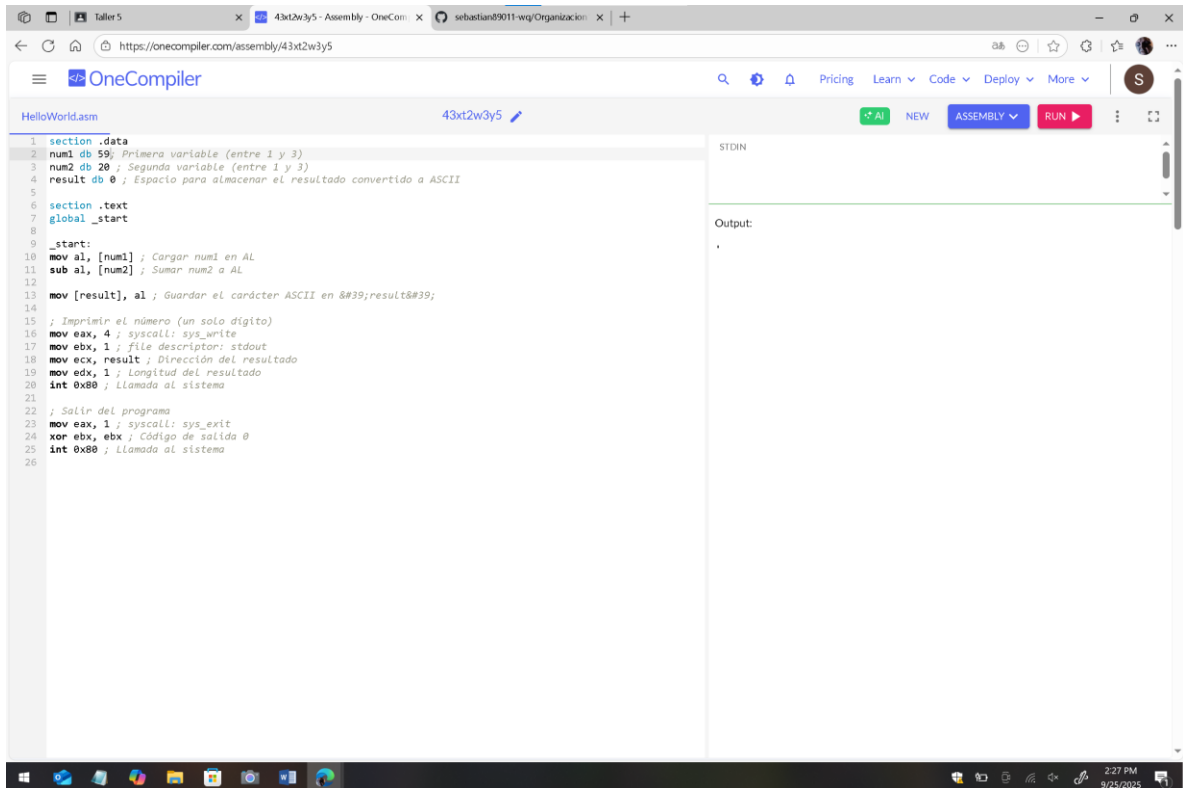
```
1 section .data
2 num1 db 63; Primera variable (entre 1 y 3)
3 num2 db 20; Segunda variable (entre 1 y 3)
4 result db 0; Espacio para almacenar el resultado convertido a ASCII
5
6 section .text
7 global _start
8
9 _start:
10 mov al, [num1]; Cargar num1 en AL
11 sub al, [num2]; Sumar num2 a AL
12
13 mov [result], al; Guardar el carácter ASCII en &#39;result&#39;
14
15 ; Imprimir el número (un solo dígito)
16 mov eax, 4; syscall: sys_write
17 mov ebx, 1; file descriptor: stdout
18 mov ecx, result; Dirección del resultado
19 mov edx, 1; Longitud del resultado
20 int 0x80; Llamada al sistema
21
22 ; Salir del programa
23 mov eax, 1; syscall: sys_exit
24 xor ebx, ebx; Código de salida 0
25 int 0x80; Llamada al sistema
26
```

STDIN

Output:

+

•



OneCompiler

43x2w3y5

```
1 section .data
2 num1 db 59; Primera variable (entre 1 y 3)
3 num2 db 20; Segunda variable (entre 1 y 3)
4 result db 0; Espacio para almacenar el resultado convertido a ASCII
5
6 section .text
7 global _start
8
9 _start:
10 mov al, [num1]; Cargar num1 en AL
11 sub al, [num2]; Sumar num2 a AL
12
13 mov [result], al; Guardar el carácter ASCII en &#39;result&#39;
14
15 ; Imprimir el número (un solo dígito)
16 mov eax, 4; syscall: sys_write
17 mov ebx, 1; file descriptor: stdout
18 mov ecx, result; Dirección del resultado
19 mov edx, 1; Longitud del resultado
20 int 0x80; Llamada al sistema
21
22 ; Salir del programa
23 mov eax, 1; syscall: sys_exit
24 xor ebx, ebx; Código de salida 0
25 int 0x80; Llamada al sistema
26
```

STDIN

Output:

+

• {

The screenshot shows the OneCompiler web interface. The browser tabs include 'Taller 5', '43xt2w3y5 - Assembly - OneCom', and 'sebastian99011-wq/Organizaci...'. The URL is 'https://onecompiler.com/assembly/43xt2w3y5'. The page title is 'HelloWorld.asm' and the assembly type is '43xt2w3y5'. The code is as follows:

```
1 section .data
2 num1 db 143; Primera variable (entre 1 y 3)
3 num2 db 20; Segunda variable (entre 1 y 3)
4 result db 0; Espacio para almacenar el resultado convertido a ASCII
5
6 section .text
7 global _start
8
9 _start:
10 mov al, [num1]; Cargar num1 en AL
11 sub al, [num2]; Sumar num2 a AL
12
13 mov [result], al; Guardar el carácter ASCII en &#39;result&#39;;
14
15 ; Imprimir el número (un solo dígito)
16 mov eax, 4; syscall: sys_write
17 mov ebx, 1; file descriptor: stdout
18 mov ecx, result; Dirección del resultado
19 mov edx, 1; Longitud del resultado
20 int 0x80; Llamada al sistema
21
22 ; Salir del programa
23 mov eax, 1; syscall: sys_exit
24 xor ebx, ebx; Código de salida 0
25 int 0x80; Llamada al sistema
26
```

The right panel shows the 'Output' section with a single line of text: '{'.

The Windows taskbar at the bottom shows the date and time as 2:28 PM on 9/25/2025.