

# **UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA**

Facultad de Ingeniería, Arquitectura y  
Diseño.

Ingeniería en Software y Tecnologías  
Emergentes

Organización de Computadoras

Taller 10



Alejandro Palomares Ceseña

### 1.-

The screenshot shows the assembly editor interface with the following assembly code:

```
1 section .data
2 char db 0
3 newline db 10
4
5 section .text
6 global _start
7
8 _start:
9
10    mov al, 0x10 ; 00010000b
11    shl al, 1 ; 00100000b = 0x20
12    or al, 0x01 ; 00100001b = 0x21
13    rol al, 1 ; 01000010b = 0x42
14    shr al, 1 ; 00100001b
15    rol al, 1 ; 01000010b
16    shr al, 1 ; 00100001b
17    shl al, 1 ; 01000010b = 'A'
18    ror al, 1 ; 00100001b
19    rol al, 1 ; 01000010b
20    sub al, 1 ; 0x41 - 'A'
21
22    mov [char], al
23
24    ; Output
25    mov eax, 4
26    mov ebx, 1
27    mov ecx, char
28    mov edx, 1
29    int 0x80
30
31    mov eax, 4
32    mov ebx, 1
33    mov ecx, newline
34    mov edx, 1
35    int 0x80
36
37    mov eax, 1
38    xor ebx, ebx
39    int 0x80
```

The output window shows the character 'A'.

### 2.-

The screenshot shows the assembly editor interface with the following assembly code:

```
1 section .data
2 char db 0
3 newline db 10
4
5 section .text
6 global _start
7
8 _start:
9
10    mov al, 0x08 ; 0000000100b
11    shl al, 3 ; 0011000000b = 0x30
12    ror al, 1 ; 0001100000b
13    rol al, 1 ; 0011000000b = 'B'
14
15    mov [char], al
16
17    ; Output
18    mov eax, 4
19    mov ebx, 1
20    mov ecx, char
21    mov edx, 1
22    int 0x80
23
24    mov eax, 4
25    mov ebx, 1
26    mov ecx, newline
27    mov edx, 1
28    int 0x80
29
30    mov eax, 1
31    xor ebx, ebx
32    int 0x80
33
34
35
```

The output window shows the character 'B'.

### 3.-

The screenshot shows the assembly editor interface with the following assembly code:

```
1 section .data
2 char db 0
3 newline db 10
4
5 section .text
6 global _start
7
8 _start:
9
10    mov al, 0x19 ; 000110001b = 25
11    shl al, 1 ; 001100010b = 0x32
12    rol al, 1 ; 01100010b = 0x62 ('b')
13    add al, 5 ; 0x67 = 'g'
14
15    mov [char], al
16
17    ; Output
18    mov eax, 4
19    mov ebx, 1
20    mov ecx, char
21    mov edx, 1
22    int 0x80
23
24    mov eax, 4
25    mov ebx, 1
26    mov ecx, newline
27    mov edx, 1
28    int 0x80
29
30    mov eax, 1
31    xor ebx, ebx
32    int 0x80
33
34
35
```

The output window shows the character 'g'.

#### 4.-

The screenshot shows a debugger interface with the following components:

- Left Panel (Assembly View):** Displays the assembly code for "HelloWorld.asm". The code includes sections for .data and .text, with various instructions like mov, shr, and int.
- Top Bar:** Includes tabs for AI, NEW, ASSEMBLY (selected), and RUN.
- STDIN:** A text input field labeled "Input for the program (Optional)" with the value "#".
- Output:** A text area showing the result of the program execution, which is currently empty.

```
1 section .data
2 char db 0
3 newline db 10
4
5 section .text
6 global _start
7
8 _start:
9
10    mov al, 0xBF ; 00001111b
11    shl al, 2 ; 00111100b = 0x3C
12    ror al, 1 ; 00011110b
13    rol al, 1 ; 00111100b
14    add al, 1 ; 0x3D = 'a'
15
16    mov [char], al
17
18 ; Output
19    mov eax, 4
20    mov ebx, 1
21    mov ecx, char
22    mov edx, 1
23    int 0x80
24
25    mov eax, 4
26    mov ebx, 1
27    mov ecx, newline
28    mov edx, 1
29    int 0x80
30
31    mov eax, 1
32    xor ebx, ebx
33    int 0x80
34
35
36
```