

Solución 1: Sin subrutinas

En el registro AX, se guarda el primer factor, llamado NUM1.

En el registro CL, se guarda el segundo factor, NUM2, que irá decrementándose.

En el registro DX, se va almacenando la suma iterativa, que será el producto final.

Si alguno de los dos factores es 0, directamente $DX = 0$; sino para la multiplicación se suma a DX el valor NUM1 un total de NUM2 veces, es decir, hasta que $NUM2 = 0$.

Solución 2: Subrutina por registro y valor

Las inicializaciones son las mismas, y en la subrutina se realiza la comprobación nula de los factores y la suma iterativa para obtener el producto final, de igual manera.

Al ejecutarse **CALL MUL**, se apila la dirección de la siguiente instrucción escrita en el programa principal, realizándose estas acciones: $SP = SP - 2$; $[SP] = [IP]$; $IP = 3000$.

Al ejecutarse **RET**, se desapila dicha dirección, realizándose: $IP = [SP]$, $SP = SP + 2$.

Solución 3: Subrutina por registro y referencia

En el registro AX, se guarda la **dirección** del primer factor, OFFSET NUM1.

En el registro CX, se guarda la dirección del segundo factor, OFFSET NUM2.

En el registro DX, nuevamente se guarda la suma iterativa, inicialmente en 0.

Dentro de la subrutina, los valores de los factores se obtienen por MDD indirecto, siempre mediante [BX]. Ya no aceptan factores con valor nulo. Primero, se obtiene NUM1 y se lo suma a DX. Luego, **se apila** este registro para poder obtener NUM2, que será decrementado, como en los casos anteriores. Una vez hecho esto, se **desapila** la suma en el registro original DX para continuar con la iteración, hasta que $NUM2 = 0$.

Al ejecutarse PUSH DX :	$SP = SP - 2$	y luego,	$[SP] = DX$
Al ejecutarse POP DX :	$DX = [SP]$	y luego,	$SP = SP + 2$

Solución 4: Subrutina por pila

Los factores se pasan por valor, y una variable “resultado” se pasa por referencia, mediante **apilamientos** en el programa principal. Luego se llama a la subrutina MUL.

Dentro de la subrutina, se apilan **todos** los registros, de modo que se puedan usar libremente como en la primera solución. BX se utiliza para acceder a valores de pila.