

Paso a paso:

La primera vuelta queda así:

```
; Memoria de datos
ORG 1000H
NUM1 DW 5H ; NUM1 y NUM2 deben ser mayores que cero
NUM2 DW 3H

; Memoria de Instrucciones
ORG 3000H ; Subrutina MUL
MUL: MOV DX, 0 //pongo 0 en DX
LAZO: MOV BX, AX //muevo lo q está en AX a BX (BX pasa a tener el offset de NUM1)-
      ADD DX, [BX] //sumo a DX el contenido de BX(DX pasa a tener 0005h = 5 en decimal)
      PUSH DX //pongo el valor de DX en la pila(0005h)
      MOV BX, CX //paso CX a BX(BX pasa a tener el offset de NUM2)
      MOV DX, [BX] //muevo el contenido de BX a DX(DX pasa a tener 0003h)
      DEC DX //decremento DX(ahora tiene 0002h)
      MOV [BX], DX //muevo a [BX] DX(BX pasa a tener 0002h, NUM2 ahora vale 0002h)
      POP DX //Recupero el valor de la cima de la pila en DX, ahora DX vale 0005h
      JNZ LAZO //como el último valor calculado de la operación q afecta los flags
              //(en este caso DEC DX)no es 0, vuelvo a LAZO

      RET

ORG 2000H ; Programa principal
      MOV AX, OFFSET NUM1 // pongo en AX la dirección (q va junto con su contenido) de NUM1
      MOV CX, OFFSET NUM2 // pongo en CX la dirección (q va junto con su contenido) de NUM2
      CALL MUL // voy a la subrutina MUL
      HLT
END
```

La segunda vuelta:

```
LAZO: MOV BX, AX //muevo lo q está en AX a BX (BX pasa a tener el offset de NUM1)-
      ADD DX, [BX] //sumo a DX el contenido de BX(DX valía 0005h, le sumo 0005h, el resultado
              //es 000ah = 10 en decimal)
      PUSH DX //pongo el valor de DX en la pila (000ah)
      MOV BX, CX //paso CX a BX (BX pasa a tener el offset de NUM2)
      MOV DX, [BX] //muevo el contenido de BX a DX (DX pasa a tener 0002h)
      DEC DX //decremento DX(ahora tiene 0001h)
      MOV [BX], DX //muevo a [BX] DX(BX pasa a tener 0001h, NUM2 ahora vale 0001h)
      POP DX //Recupero el valor de la cima de la pila en DX, ahora DX vale 000ah
      JNZ LAZO //como el último valor calculado de la operación q afecta los flags
              //(en este caso DEC DX)no es 0, vuelvo a LAZO
```

La tercera vuelta:

```
LAZO: MOV BX, AX //muevo lo q está en AX a BX (BX pasa a tener el offset de NUM1)-
      ADD DX, [BX] //sumo a DX el contenido de BX(DX valía 000ah, le sumo 0005h, el resultado
              //es 000fh = 15 en decimal)
      PUSH DX //pongo el valor de DX en la pila (000fh)
      MOV BX, CX //paso CX a BX (BX pasa a tener el offset de NUM2)
      MOV DX, [BX] //muevo el contenido de BX a DX (DX pasa a tener 0001h)
      DEC DX //decremento DX(ahora tiene 0000h)
      MOV [BX], DX //muevo a [BX] DX(BX pasa a tener 0000h, NUM2 ahora vale 0000h)
      POP DX //Recupero el valor de la cima de la pila en DX, ahora DX vale 000fh
      JNZ LAZO //como el último valor calculado de la operación q afecta los flags
              //(en este caso DEC DX)es 0, no vuelvo a LAZO

      RET // Vuelvo a la rutina principal.
              // DX (15) tiene el resultado de la multiplicación de NUM1 (5) y NUM2 (3)
```