

Ayuda para la resolución del primer problema del trabajo práctico Repaso de Arquitectura.

El código ensamblado desde la dirección de memoria CS:0100 hasta la dirección de memoria CS:0112 cuando se ejecuta permite ingresar una secuencia de bytes (byte por byte) mientras no se tipee el byte "\$".

Probar la ejecución del siguiente código ingresando una secuencia de bytes finalizando con el carácter \$.

Probar la ejecución del siguiente código ingresando byte por byte y tipeando un ENTER para cada byte.

Ver la diferencia entre ambos tipos de ingreso con el comando E 2000

La siguiente imagen muestra el desensamblado del código

```

-R
AX=0124 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=2002 DI=0000
DS=0C9B ES=0C9B SS=0C9B CS=0C9B IP=0100 NU UP EI PL ZR NA PE MC
0C9B:0100 BE0020      MOV     SI,2000
-U 0100 0110
0C9B:0100 BE0020      MOV     SI,2000
0C9B:0103 B401      MOV     AH,01
0C9B:0105 CD21      INT     21
0C9B:0107 8804      MOV     [SI],AL
0C9B:0109 3C24      CMP     AL,24
0C9B:010B 7403      JZ      0110
0C9B:010D 46        INC     SI
0C9B:010E EBF3      JMP     0103
0C9B:0110 CD20      INT     20

```

Explicación del código

Código	Descripción
MOV SI,2000	Transfiere la constante 2000 al registro SI
MOV AH,01	Transfiere la constante 01 a la parte alta del registro AX, esta es la función para aceptar un carácter desde el teclado
INT 21	Interrupción que resuelve el ingreso del carácter por teclado
MOV [SI],AL	Transfiere el contenido de la parte baja del registro AX a la dirección de memoria registrada en el registro SI
CMP AL,24	Compara el contenido de AL con \$
JZ 0110	Si el contenido de AL es 24, se bifurca a la dirección de memoria CS:0110 para ejecutar la instrucción INT 20
INC SI	Incrementa en 1 el registro SI
JMP 0103	Bifurca la ejecución del programa a la

	dirección CS:0103
INT 20	Interrupción para finalizar el programa

Esta imagen muestra una ejecución del programa utilizando el comando G y tipeando un ENTER después de cada byte ingresado.

```

-R
AX=0124 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=2002 DI=0000
DS=0C9B ES=0C9B SS=0C9B CS=0C9B IP=0100 NU UP EI PL ZR NA PE NC
0C9B:0100 BE0020      MOV     SI,2000
-U 0100 0110
0C9B:0100 BE0020      MOV     SI,2000
0C9B:0103 B401      MOV     AH,01
0C9B:0105 CD21      INT     21
0C9B:0107 8804      MOV     [SI],AL
0C9B:0109 3C24      CMP     AL,24
0C9B:010B 7403      JZ      0110
0C9B:010D 46      INC     SI
0C9B:010E EBF3      JMP     0103
0C9B:0110 CD20      INT     20

-G
$
El programa ha terminado de forma normal
    
```

Esta imagen muestra el almacenamiento en memoria a partir de la dirección DS:2000 de los bytes ingresados por teclado. Se puede ver que después de tipear la letra L, se tipearon dos ENTER.

```

C:\WINDOWS\system32\debug.exe
-R
AX=0124 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=2002 DI=0000
DS=0C9B ES=0C9B SS=0C9B CS=0C9B IP=0100 NU UP EI PL ZR NA PE NC
0C9B:0100 BE0020      MOV     SI,2000
-D 2000
0C9B:2000 48 0D 4F 0D 4C 0D 0D 41-0D 20 0D 4D 0D 55 0D 4E  H.O.L.A. .M.U.N
0C9B:2010 0D 44 0D 4F 0D 24 72 69-62 61 20 53 45 54 20 73  .D.O.$riba SET s
0C9B:2020 69 6E 20 6E 69 6E 67 A3-6E 20 70 61 72 A0 6D 65  in ning.n par.me
0C9B:2030 74 72 6F 20 70 61 72 61-20 76 65 72 20 6C 61 73  tro para ver las
0C9B:2040 20 76 61 72 69 61 62 6C-65 73 20 64 65 20 65 6E  variables de en
0C9B:2050 74 6F 72 6E 6F 20 61 63-74 75 61 6C 65 73 2E 0D  torno actuales..
0C9B:2060 0A 3B 4D 75 65 73 74 72-61 20 6F 20 65 73 74 61  .;Muestra o esta
0C9B:2070 62 6C 65 63 65 20 6C 61-20 68 6F 72 61 20 64 65  blece la hora de
    
```

El código del programa esta almacenado a partir de la dirección de memoria CS:0100 y demando 18 bytes de memoria

En hexadecimal es el siguiente código

BE0020B401CD2188043C24740346EBF3CD20

En lenguaje de máquina el código es el siguiente

```
1011111000000000001000001011010000000001110011010010000110001000000001
0000111100001001000111010000000011010001101110101111110011110011010010
0000
```

Convertir el carácter ingresado almacenado como su código ASCII en su número equivalente

La siguiente imagen muestra un programa que pide un byte por teclado y lo almacena en la dirección de memoria DS:2000, en el ejemplo se ingresa el carácter 1 y se ve como en memoria se almacena el código ASCII 31

```

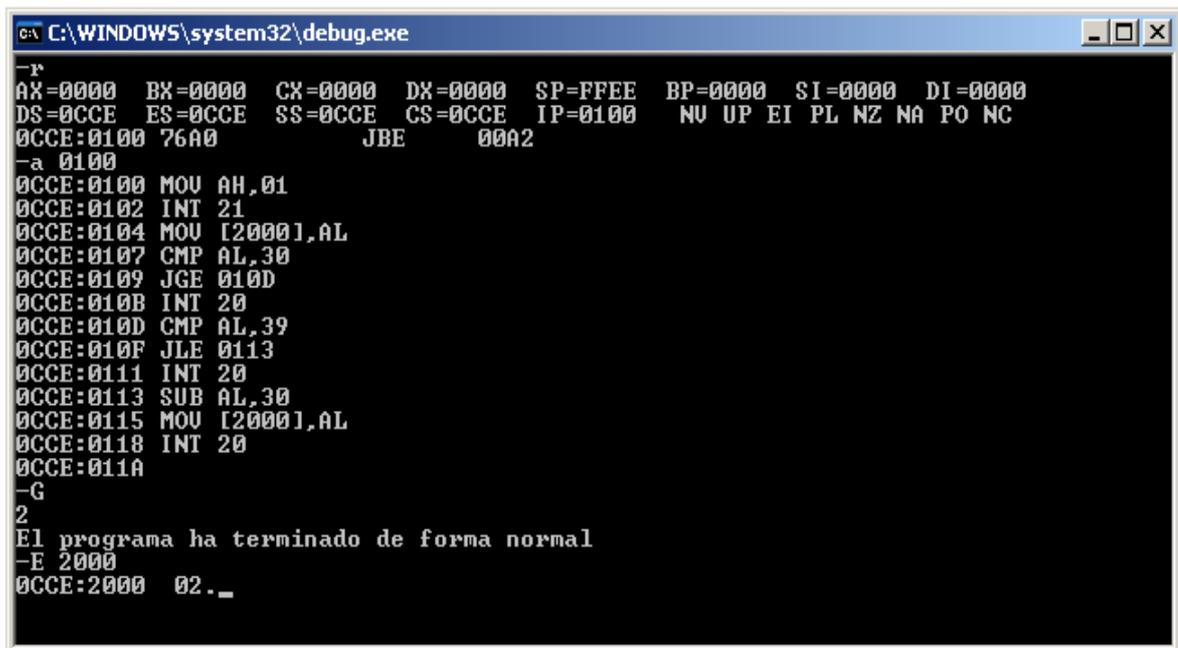
-r
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0000 ES=0000 SS=0000 CS=0000 IP=0100  NU UP EI PL NZ NA PO NC
0CCC:0100 B401          MOV     AH,01
-a 0100
0CCC:0100 mov ah,01
0CCC:0102 int 21
0CCC:0104 mov [2000],al
0CCC:0107 int 20
0CCC:0109
-g
1
El programa ha terminado de forma normal
-d 2000
0CCC:2000 31 65 73 20 2E 0D 0A 0D-0A 43 41 4C 4C 20 5B 75  les .....CALL [u
0CCC:2010 6E 69 64 61 64 3A 5D 5B-72 75 74 61 5D 61 72 63  nidad:][rutalarc
0CCC:2020 68 69 76 6F 20 5B 70 61-72 A0 6D 65 74 72 6F 73  hivo [par.metros
0CCC:2030 5D 0D 0A 0D 0A 7F 20 20-70 61 72 A0 6D 65 74 72  l..... par.metr
0CCC:2040 6F 73 20 20 20 45 73 70-65 63 69 66 69 63 61 20  os Especifica
0CCC:2050 63 75 61 6C 71 75 69 65-72 20 69 6E 66 6F 72 6D  cualquier inform
0CCC:2060 61 63 69 A2 6E 20 64 65-20 6C 61 20 6C A1 6E 65  acion de la l.ne
0CCC:2070 61 20 64 65 20 63 6F 6D-61 6E 64 6F 73 0D 0A 20  a de comandos..

```

Para operar aritméticamente con el dato ingresado tenemos que convertir el código ASCII 31 al valor 1

La siguiente figura completa el programa anterior convirtiendo el código ASCII almacenado en la dirección DS:2000 a su número equivalente y dejándolo en la misma dirección de memoria.

Muestra como ingresamos el carácter 2 y queda almacenado en la dirección DS:2000 como el número 2.



```
C:\WINDOWS\system32\debug.exe
-r
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0CCE ES=0CCE SS=0CCE CS=0CCE IP=0100  NU UP EI PL NZ NA PO NC
0CCE:0100 76A0          JBE     00A2
-a 0100
0CCE:0100 MOV AH,01
0CCE:0102 INT 21
0CCE:0104 MOV [2000],AL
0CCE:0107 CMP AL,30
0CCE:0109 JGE 010D
0CCE:010B INT 20
0CCE:010D CMP AL,39
0CCE:010F JLE 0113
0CCE:0111 INT 20
0CCE:0113 SUB AL,30
0CCE:0115 MOV [2000],AL
0CCE:0118 INT 20
0CCE:011A
-G
2
El programa ha terminado de forma normal
-E 2000
0CCE:2000 02._
```