



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO
CENTRO UNIVERSITARIO UAEM ATLACOMULCO



Unidad de aprendizaje: Lenguaje Ensamblador

Docente: M. en C. C. Juan Carlos Ambriz Polo

Evaluación: Segundo parcial

Fecha de asignación: 7/05/2019

Fecha límite de entrega: 28/05/2019

Alumno: Gustavo Blas Duran

Nota: Recuerde que las primeras entregas obtiene el 100% del valor, posteriormente por cada clase que transcurra reduce el 10% esto hasta llegar a la fecha límite, ya que después de dicha fecha no se aceptan entregas. Lea detenidamente las instrucciones y evite omitir pasos, trabaje sobre este formato de lo contrario no se recibirá la práctica.

Practica 3 Instrucciones aritméticas

Objetivo:

Implementar ciclos numéricos así como saltos y saltos condicionales en lenguaje ensamblador.

Actividad 1: Resumir conceptos principales del tema

Redacte un resumen de operadores aritméticos, así como de nemónicos para realizar ajustes necesarios que permitan almacenar y desplegar resultados con unidades y decenas (es decir de dos cifras). Nota: extensión mínima dos cuartillas.

Las instrucciones ADD y SUB realizan sumas y restas sencillas de datos binarios. Los números binarios negativos están representados en la forma de complemento a dos: se invierte todos los bits del número positivo y se suma 1. Como con otras instrucciones, no existen operaciones directas de memoria a memoria.

En las computadoras el formato natural para la aritmética es el binario. Pues este no causa mayores problemas, siempre y cuando el programa defina sus datos. Sin embargo, para muchos propósitos, los datos numéricos se introducen desde el teclado como caracteres ASCII, en formato de base 10, de manera similar, el despliegue de valores numéricos en la pantalla es en formato ASCII. Un formato relacionado, decimal codificado en binario (BCD), tiene uso ocasional y aparece como desempaqueado y empaquetado. La PC proporciona varias instrucciones que facilitan la aritmética sencilla y la conversión de datos ASCII a binario para aplicar la aritmética, así como las técnicas para convertir los resultados binarios de regreso a formato ASCII para su visualización.

AAA ajusta ASCII después de sumar, AAS después de restar, AAM después de multiplicar, AAD para dividir, DAA ajuste decimal después de sumar y DAS ajuste decimal después de restar.

Datos en formato decimal: el sistema de la PC también puede usar formato decimal codificado en binario que facilita algunas operaciones aritméticas limitadas. Dos usos del formato BCD son, 1- el BCD permite el redondeo apropiado de números sin pérdida de precisión, una característica que es particularmente útil para el manejo de cantidades monetarias (pesos y centavos). El redondeo de numero binarios que representan pesos y centavos puede provocar una perdida en la precisión. 2- con frecuencia es más sencillo realizar aritmética con números pequeños introducidos desde el teclado o que son escritos en la pantalla o en la impresora.

Se pueden almacenar datos BCD como desempquetado y empaquetado: BCD desempquetado tiene un solo dígito BCD en los cuatro bits inferiores de cada byte, con ceros en los cuatro bits superiores. El formato ASCII también es desempquetado, aunque no se le llaman así.

EL BCD empaquetado contiene dígitos BCD, uno de los cuatro bits superiores y en uno de los cuatro bits inferiores. Este formato es muy común para la aritmética que utiliza coprocesador numérico. Definido como 10 bytes con la directiva DT. El procesador realiza aritmética en valores ASCII y BCD un dígito a la vez.

Ya que los datos que se ingresan desde el teclado están en formato ASCII, la representación en memoria del numero de un numero decimal ingresado tal como 1234 es 31323334H. pero realizar la aritmética sobre el numero decimal ingresado implica un tratamiento especial. Las instrucciones AAA y AAS realizan aritmética de manera directa sobre números ASCII.

La instrucción MUL (multiplicación sin signo) viene en tres versiones; la primera multiplica un operando de 8 bits por al, la segunda multiplica un operando de 16 bits por AX y la tercera multiplican un operando de 32 bits por EAX. El multiplicador y el multiplicando son del mismo tamaño, y el producto es del doble de su tamaño. Los tres formatos aceptan operandos de registro y de memoria; pero no operandos inmediatos.

El operando individual es el multiplicador. Como el operando de destino es del doble del tamaño del multiplicando y del multiplicador, no puede ocurrir un desbordamiento. MUL activa las banderas de Acarreo y desbordamiento si la mitad superior del producto no es igual a cero. Por lo general, la bandera Acarreo se utiliza para la aritmética sin signo, por lo que aquí nos enfocaremos en eso. Por ejemplo ¿, cuando AX se multiplica por un operando de 16 bits, el producto se almacena en DX:AX. La bandera de acarreo se activa si DX no es igual a cero.

La instrucción IMUL (multiplicación con signo) realiza la multiplicación de enteros sin signo, preservando el signo del producto. EL conjunto de Instrucciones IA.32 soporta tres formatos para esta instrucción: un operando, dos operandos y tres operandos. En el formato de un operando, el multiplicador y el multiplicando son del mismo tamaño y el producto es el doble de su tamaño.

Como en el caso de MUL, el tamaño de almacenamiento del producto hace que el desbordamiento sea imposible en la instrucción IMUL de un operando. Las banderas de acarreo y desbordamiento se activan si a la mitad superior del producto no es una extensión del signo de la mitad inferior. Se puede utilizar esta información para decidir si se debe ignorar o no la mitad superior del producto.

El formato de dos operandos de esta instrucción almacena el producto en el primer operando, el primer operando debe ser un registro, el segundo operando puede ser un registro, operando de memoria o valor inmediato.

La instrucción DIV (división sin signo) realiza la división de enteros con signo de 8 bits, 16 bits y 32 bits. El registro individual u operando de memoria es el divisor. En la versión de división de 8 bits el dividendo se encuentra en AX, el divisor registro en memoria de 8 bits, el cociente en AL, y el residuo en AH.

La división de enteros con signo es casi idéntica a la división sin signo, con una importante diferencia: el dividendo implicado debe tener una extensión completa del signo antes de realizar la división. Primero veremos las instrucciones para la extensión del signo. Después las aplicaremos a la instrucción de división de enteros con signo, IDIV.

La instrucción CBW (convierte byte a palabra) extiende el bit de AL hacia AH, preservando el signo del número. La instrucción CWD (convertir palabra a doble palabra) extiende el bit del signo de AX hacia DX. La instrucción CDQ (convertir doble palabra a palabra cuádruple) extiende el bit de signo EAX hacia EDX.

La instrucción IDIV (división con signo) realiza una división de enteros con signo, usando los mismos operados que DIV. Antes de ejecutar la división de 8 bits, se debe extender por completo el signo del dividendo (AX). El residuo siempre tiene el mismo signo que el dividendo.

Actividad 2: Codificación de programas en ensamblador

Instrucciones: Codifique los siguientes programas en ensamblador

1. Programa en ensamblador que despliegue un menú como el siguiente, realice la función indicada y repita el menú hasta que se elija la opción salir, para cada opción del menú se permitirá ingresar nuevos datos por el usuario

1.- Suma

2.- Resta

3.- Multiplicación

4.- División

5.-Salir

```
1  .model small
2  JUMPS
3  .stack
4  .data
5
6      datosMenu db 10,13,"    MENU",10,13,"1.-Suma",10,13,"2.-
Resta",10,13,"3.-Multiplicacion",10,13,"4.-Division",10,13,"5.-
Salir",10,13,"Inserte opcion: $"
7      msgDefault db 10,13,"Error... no has insertado un caracter
invalido. intenta de nuevo ",10,13,36
8      msgTresDigitos db 10,13,"Resultado >= 100. No podemos
imprimir",36
9      msgNumNeg db 10,13,"El segundo numero es mayor al primero, el
resultado es negativo",36
10     instruccion db 10,13,"inserta operacion:  ",36
11     sleep db 10,13,10,13,10,13,"Presiona cualquier tecla para
continuar...$"
12     residuoDiv db 10,10,"Residuo: $"
13     salto db 10,13,36
14     n1 db 0
15     n2 db 0
16     n1aux db 0
17     n2aux db 0
18     pot db 10
19     ;variables para formato de impresion
20     c1 db 0
21     c2 db 0
22     c3 db 0
23     c4 db 0
24     r db 0
```

```

25  .code
26      mov ax,@data
27      mov ds,ax
28      ;cls screen
29      mov ax,0600h
30      mov bl,0ah
31      mov cx,0000h
32      mov dx,184fh
33      int 10h
34  inicio:
35      mov ah,00h
36      mov ax,3
37      int 10h
38      ;se muestra el menu
39      mov ah,09h
40      mov dx,offset datosMenu
41      int 21h
42      mov ah,01h ;se inserta caracter
43      int 21h
44      cmp al,31h ;comparacion de la variable con opciones
45      je suma
46      cmp al,32h
47      je resta
48      cmp al,33h
49      je multiplicacion
50      cmp al,34h
51      je division
52      cmp al,35h
53      je salir
54      cmp al,36h
55      jge default
56      cmp al,30h
57      jle default
58      ;inician las funciones
59      suma:
60          ;muestra mensaje
61          mov ah,09h
62          mov dx,offset instruccion
63          int 21h
64          mov ah,01h ;decena primer numero
65          int 21h
66          ;validacion de numeros
67          mov c1,al ;*****c1
68          sub al,30h
69          mul pot
70          mov n1,al
71          mov ah,01h ;unidades de primer digito
72          int 21h
73          mov c2,al ;*****c2
74          sub al,30h
75          add al,n1
76          mov n1,al
77          ;n1 contiene el valor del primer numero
78

```

```

79      mov ah,02h
80      mov dl,'+'
81      int 21h
82      mov ah,01h ;decenas segundo digito
83      int 21h
84      mov c3,al ;*****c3
85      sub al,30h
86      mul pot
87      mov n2,al
88      mov ah,01h ;unidades segundo digito
89      int 21h
90      mov c4,al ;*****c4
91      sub al,30h
92      add al,n2
93      mov n2,al ;n2 contiene el valor del segundo numero
94      ;hacer suma de n1,n2
95      mov al,n1
96      add al,n2
97      aam
98      mov n1aux,ah
99      mov n2aux,al
100     mov ah,09h ;se imprime valores con formato
101     mov dx,offset salto
102     int 21h
103     int 21h
104     int 21h
105     mov ah,02h
106     mov dl,c1
107     int 21h
108     mov dl,c2
109     int 21h
110     mov dl,'+'
111     int 21h
112     mov dl,c3
113     int 21h
114     mov dl,c4
115     int 21h
116     mov dl,'='
117     int 21h
118     mov ah,02h ;imprimir el numero
119     mov dl,n1aux;parte alta del resultado
120     add dl,30h
121     int 21h
122     mov ah,02h
123     mov dl,n2aux;parte baja del resultado
124     add dl,30h
125     int 21h
126     mov ah,09h ;se espera hasta insertar tecla
127     mov dx,offset sleep
128     int 21h
129     mov ah,01h
130     int 21h
131     jmp inicio ;regresa a inicio
132

```

```

133
134     resta:
135         ;muestra mensaje
136         mov ah,09h
137         mov dx,offset instruccion
138         int 21h
139         mov ah,01h ;decena primer numero
140         int 21h
141         mov c1,al ;*****c1
142         sub al,30h
143         mul pot
144         mov n1,al
145         mov ah,01h ;unidades de primer digito
146         int 21h
147         mov c2,al ;*****c2
148         sub al,30h
149         add al,n1
150         mov n1,al ;n1 contiene el valor del primer numero
151         mov ah,02h
152         mov dl,'-'
153         int 21h
154         mov ah,01h ;decenas segundo digito
155         int 21h
156         mov c3,al ;*****c3
157         sub al,30h
158         mul pot
159         mov n2,al
160         mov ah,01h ;unidades segundo digito
161         int 21h
162         mov c4,al ;*****c4
163         sub al,30h
164         add al,n2
165         mov n2,al ;n2 contiene el valor del segundo numero
166         ;compara valor de digitos
167         mov al,n1
168         cmp n2,al
169         jg numNeg
170         ;hacer resta de n1,n2
171         mov al,n1
172         sub al,n2
173         aam
174         mov n1aux,ah
175         mov n2aux,al
176         mov ah,09h
177         mov dx,offset salto
178         int 21h
179         int 21h
180         int 21h
181         mov ah,02h ;s imprime en formato
182         mov dl,c1
183         int 21h
184         mov dl,c2
185         int 21h
186         mov dl,'-'
187         int 21h
188         mov dl,c3
189         int 21h

```

```

190     mov dl,c4
191     int 21h
192     mov dl,'='
193     int 21h
194     ;imprimir el numero
195     mov ah,02h
196     mov dl,n1aux
197     add dl,30h
198     int 21h
199     mov ah,02h
200     mov dl,n2aux
201     add dl,30h
202     int 21h
203     mov ah,09h
204     mov dx,offset sleep
205     int 21h
206     mov ah,01h
207     int 21h
208     jmp inicio ;regresa a inicio
209     numNeg:    ;si el resultado es un numero negativo
210         mov ah,09h
211         mov dx,offset msgNumNeg
212         int 21h
213         mov ah,09h
214         mov dx,offset sleep
215         int 21h
216         mov ah,01h
217         int 21h
218         jmp inicio
219
220     multiplicacion:
221         ;muestra mensaje
222         mov ah,09h
223         mov dx,offset instruccion
224         int 21h
225         mov ah,01h ;decena primer numero
226         int 21h
227         mov c1,a1 ;*****c1
228         sub al,30h
229         mul pot
230         mov n1,a1
231         mov ah,01h ;unidades de primer digito
232         int 21h
233         mov c2,a1 ;*****c2
234         sub al,30h
235         add al,n1
236         mov n1,a1 ;n1 contiene el valor del primer numero
237         mov ah,02h
238         mov dl,'*'
239         int 21h
240         mov ah,01h ;decenas segundo digito
241         int 21h
242         mov c3,a1 ;*****c3
243         sub al,30h
244         mul pot
245         mov n2,a1
246         mov ah,01h ;unidades segundo digito

```



```

247     int 21h
248     mov c4,al                ;*****c4
249     sub al,30h
250     add al,n2
251     mov n2,al    ;n2 contiene el valor del segundo numero
252     ;hacer multiplicacion de n1,n2
253     mov al,n1
254     mov cl,n2
255     mul cl
256     aam
257     mov n1aux,ah
258
259     mov n2aux,al
260     mov ah,09h
261     mov dx,offset salto
262     int 21h
263     int 21h
264     int 21h
265     mov ah,02h
266     mov dl,c1
267     int 21h
268     mov dl,c2
269     int 21h
270     mov dl,'*'
271     int 21h
272     mov dl,c3
273     int 21h
274     mov dl,c4
275     int 21h
276     mov dl,'='
277     int 21h
278     ;imprimir el numero
279     mov ah,02h
280     mov dl,n1aux
281     add dl,30h
282     int 21h
283     mov ah,02h
284     mov dl,n2aux
285     add dl,30h
286     int 21h
287     mov ah,09h
288     mov dx,offset sleep
289     int 21h
290     mov ah,01h
291     int 21h
292     jmp inicio    ;regresa a inicio
293
294 division:
295     ;muestra mensaje
296     mov ah,09h
297     mov dx,offset instruccion
298     int 21h
299     mov ah,01h    ;decena primer numero
300     int 21h
301     mov c1,al                ;*****c1
302     sub al,30h
303     mul pot

```

```

304     mov n1,al
305     mov ah,01h ;unidades de primer digito
306     int 21h
307     mov c2,al ;*****c2
308     sub al,30h
309     add al,n1
310     mov n1,al ;n1 contiene el valor del primer numero
311     mov ah,02h
312     mov dl,246
313     int 21h
314     mov ah,01h ;decenas segundo digito
315     int 21h
316     mov c3,al ;*****c3
317     sub al,30h
318     mul pot
319     mov n2,al
320     mov ah,01h ;unidades segundo digito
321     int 21h
322     mov c4,al ;*****c4
323     sub al,30h
324     add al,n2
325     mov n2,al ;n2 contiene el valor del segundo numero
326     ;hacer division de n1,n2
327     xor ax,ax
328     mov cl,n1 ;dividendo
329     add ax,cx
330     mov bl,n2 ;divisor
331     div n2
332     aam
333     mov n1aux,ah ;residuo
334     mov n2aux,al ;cociente
335     aad
336     mov ah,09h
337     mov dx,offset salto
338     int 21h
339     int 21h
340     int 21h
341     mov ah,02h
342     mov dl,c1
343     int 21h
344     mov dl,c2
345     int 21h
346     mov dl,246
347     int 21h
348     mov dl,c3
349     int 21h
350     mov dl,c4
351     int 21h
352     mov dl,'='
353     int 21h
354     ;imprimir el numero
355     mov al,n1aux
356     aam
357     mov bx,ax
358     mov ah,02h
359     mov dl,bl
360     add dl,30h

```

```

361         int 21h
362         mov al,n2aux
363         aam
364         mov bx,ax
365         mov ah,02h
366         mov dl,bl
367         add dl,30h
368         int 21h
369
370         mov ah,09h
371         mov dx,offset sleep
372         int 21h
373         mov ah,01h
374         int 21h
375         jmp inicio ;regresa a inicio
376
377     mayor100:
378         mov ah,09h
379         mov dx,offset msgTresDigitos
380         int 21h
381         mov ah,09h
382         mov dx,offset sleep
383         int 21h
384         mov ah,01h
385         int 21h
386         jmp inicio
387
388     default:
389         mov ah,00h
390         mov ax,3
391         int 10h
392         jmp inicio
393
394     salir: ;devuelve el control al DOS
395         mov ah,4ch
396         int 21h
397
398 end

```

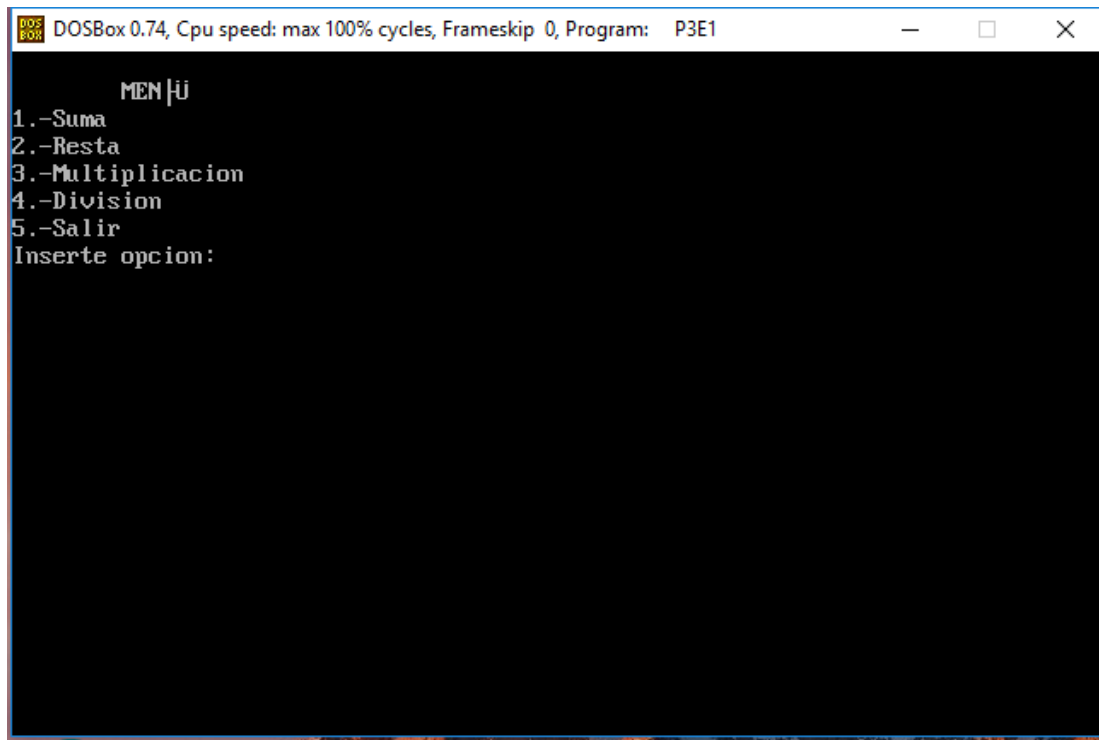


Ilustración 1 inicio de programa

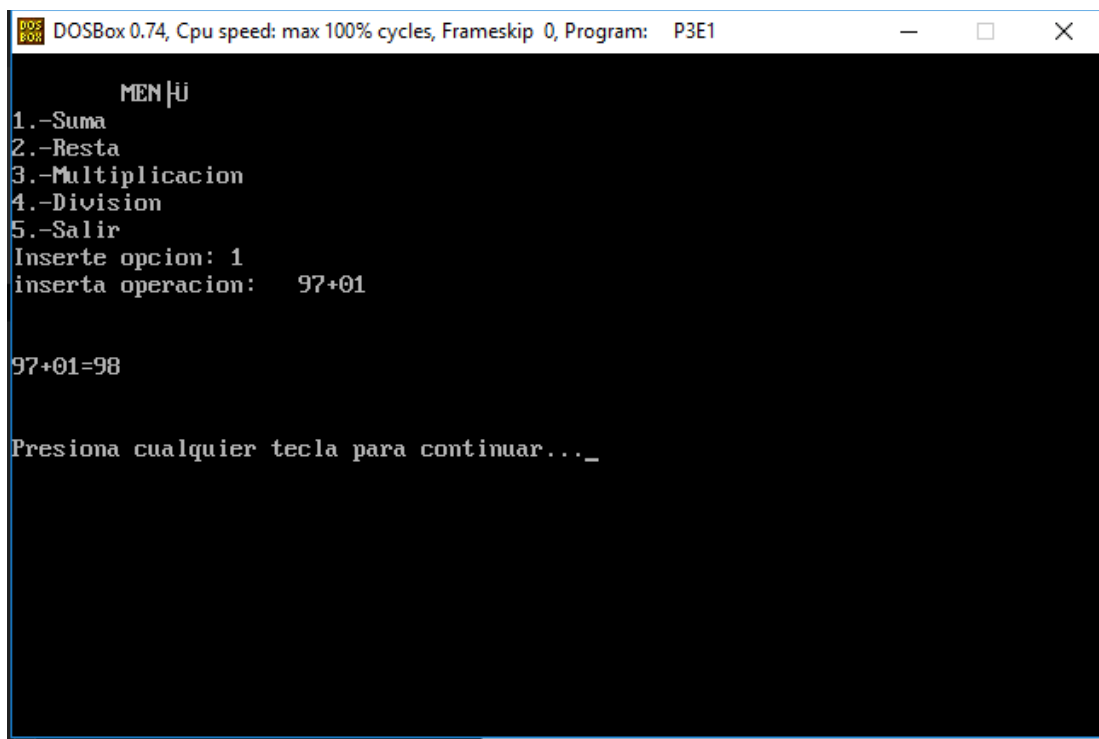
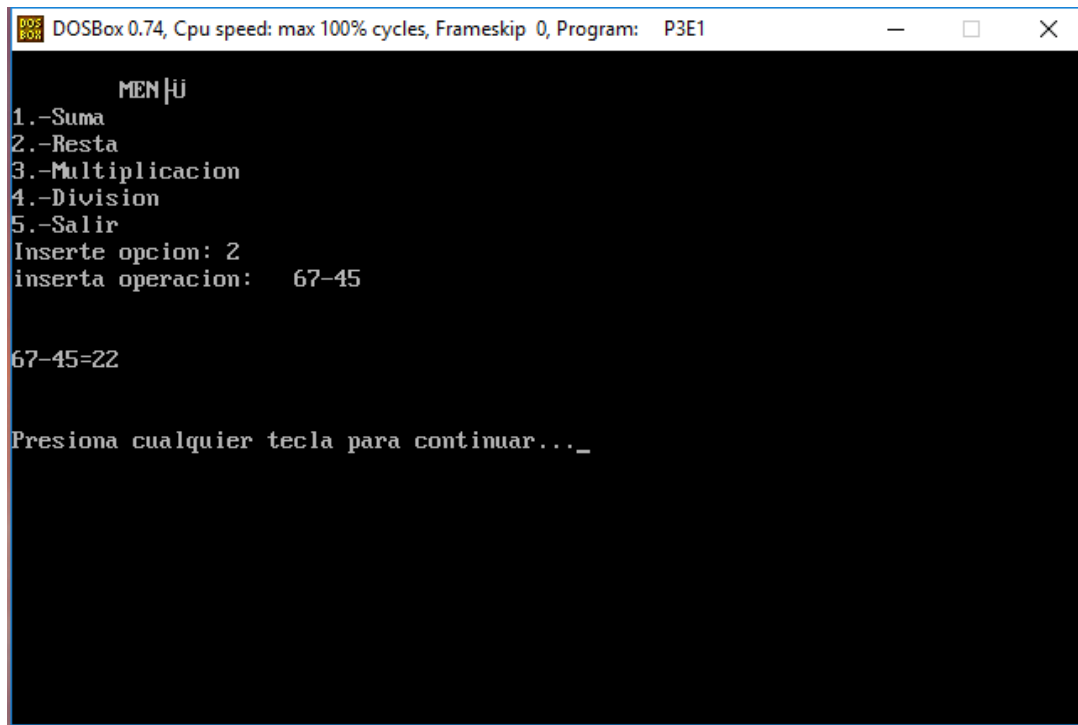


Ilustración 2 ejecución de suma de números



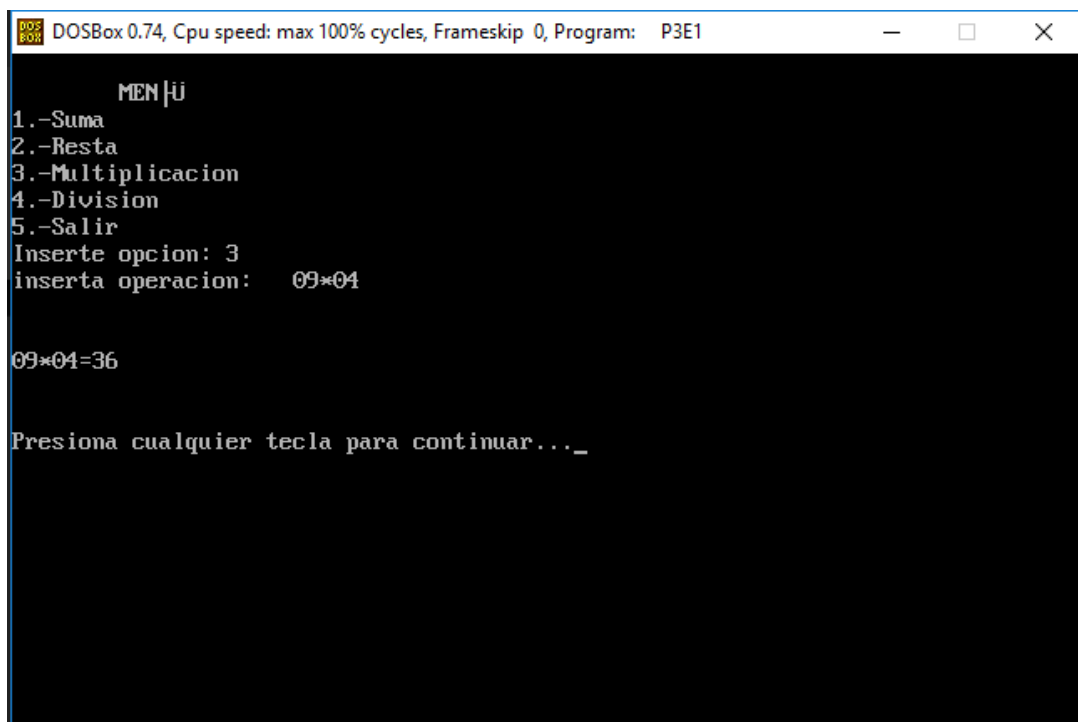
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: P3E1

```
MEN |U
1.-Suma
2.-Resta
3.-Multiplicacion
4.-Division
5.-Salir
Inserte opcion: 2
inserta operacion: 67-45

67-45=22

Presiona cualquier tecla para continuar..._
```

Ilustración 3 ejecución de resta de números



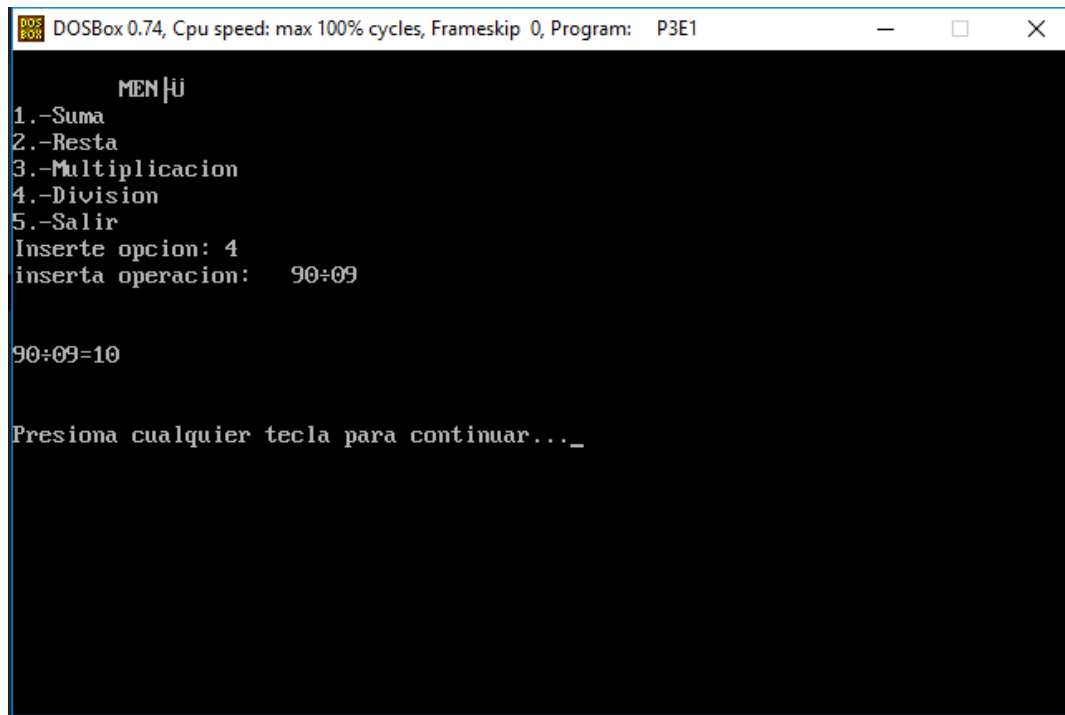
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: P3E1

```
MEN |U
1.-Suma
2.-Resta
3.-Multiplicacion
4.-Division
5.-Salir
Inserte opcion: 3
inserta operacion: 09*04

09*04=36

Presiona cualquier tecla para continuar..._
```

Ilustración 4 ejecución de multiplicación de números



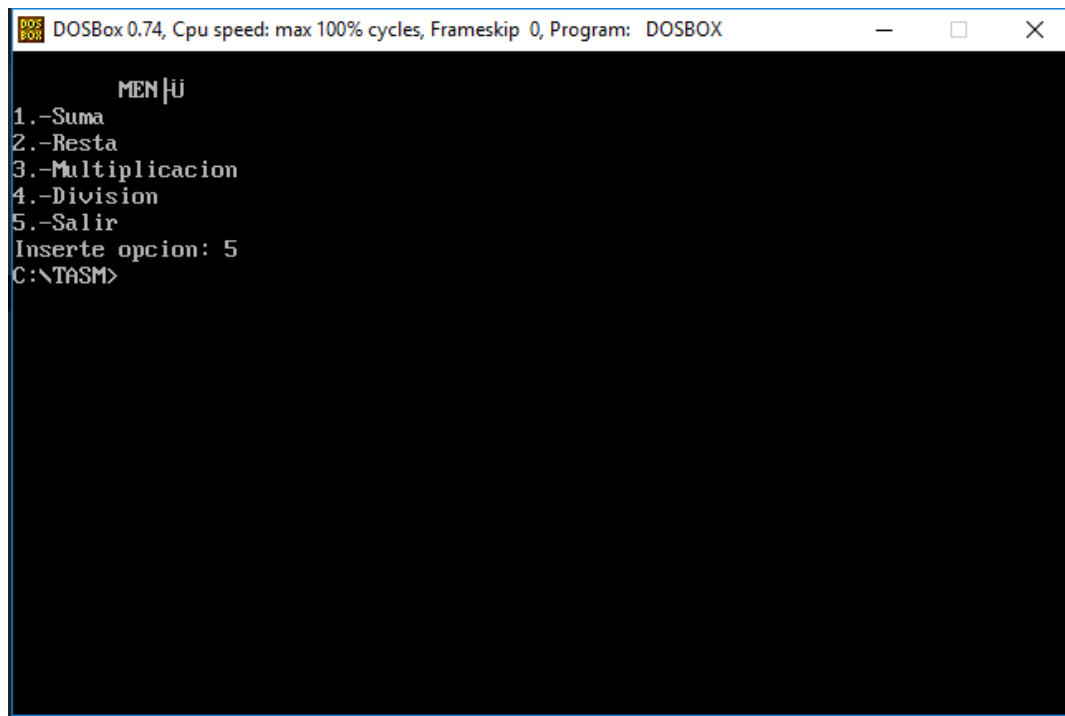
```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: P3E1

MENÚ
1.-Suma
2.-Resta
3.-Multiplicacion
4.-Division
5.-Salir
Inserte opcion: 4
inserta operacion: 90÷09

90÷09=10

Presiona cualquier tecla para continuar..._
```

Ilustración 5 ejecución de división de números



```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: DOSBOX

MENÚ
1.-Suma
2.-Resta
3.-Multiplicacion
4.-Division
5.-Salir
Inserte opcion: 5
C:\TASM>
```

Ilustración 6 opción salir del programa.

2. Codifique un programa en lenguaje ensamblador que permita genera la factorial de un número, considere como limite la impresión del resultado con al menos 2 cifras.

```
1  .model small
2  .stack
3  .data
4      jumps
5      msgInicio db 10,13,"Factorial de un numero , solo hasta el
numero 4", 36
6      msgInsertar db 10,13,"Inserta 1-2-3-4: ",36
7      msgResultado db 10,13,"Resultado Final: ", 36
8      msgValidarNumero db 10,13,"Solo numeros del 1-4...",36
9      saltodelinea db 10,13,36
10     res db 1
11     factorial db 0
12 .code
13     mov ax,@data
14     mov ds,ax
15     ;se limpia la pantalla
16     mov ah,00h
17     mov ax,3
18     int 10h
19     mov ah,09h
20     mov dx,offset msgInicio
21     int 21h
22     insertar:
23     mov ah,09h
24     mov dx,offset saltodelinea
25     int 21h
26     int 21h
27     mov al,0
28     mov dx,offset msgInsertar
29     int 21h
30     mov ah,01h ;se pide el numero al usuario
31     int 21h
32
33     cmp al,30h
34     jl  validarNumeros;menor
35     cmp al,34h
36     jg  validarNumeros;mayor
37
38     sub al,30h
39     mov cl,al
40     mov factorial,al
41     mov ah,02h
42     mov dl,'!'
43     int 21h
44     mov ah,09h
45     mov dx,offset saltodelinea
46     int 21h
```

```

47     int 21h
48
49     ciclo:
50         mov al,res    ;operador a multiplicar
51         mov bl,cl
52         mul bl        ;se realiza la multiplicacion
53         mov res,al    ;se respalda el resultado en res
54         mov al,res
55         aam           ;ajuste ASCII despues de la multiplicacion
56         mov bx,ax
57         mov ah,02h
58         mov dl,bh     ;se imprimen los numeros
59         add dl,30h
60         int 21h
61         mov ah,02h
62         mov dl,bl     ;parte naja del resultado
63         add dl,30h
64         int 21h
65         mov ah,02h
66         mov dl,' '
67         int 21h
68     loop ciclo
69     ;se imprimen en formato
70     mov ah,09h
71     mov dx,offset saltodelinea
72     int 21h
73     mov dx,offset msgResultado
74     int 21h
75     mov ah,02h
76     mov dl,factorial
77     add dl,30h
78     int 21h
79     mov ah,02h
80     mov dl,'!'
81     int 21h
82     mov dl,'='
83     int 21h
84     mov al,res
85     aam               ;ajuste ASCII para multiplicacion
86     mov bx,ax
87     mov ah,02h
88     mov dl,bh        ;parte alta
89     add dl,30h
90     int 21h
91     mov ah,02h
92     mov dl,bl        ;parte baja
93     add dl,30h
94     int 21h
95     mov ah,09h
96     mov dx,offset saltodelinea
97     int 21h
98     int 21h
99     ;devuelve el control al DOS
100    mov ah,4ch
101    int 21h
102
103    validarNumeros:

```



```
104      mov ah,00h
105      mov ax,3
106      int 10h
107      mov ah,09h
108      mov dx,offset msgValidarNumero
109      int 21h
110      jmp insertar
111
112
113
114  end
```

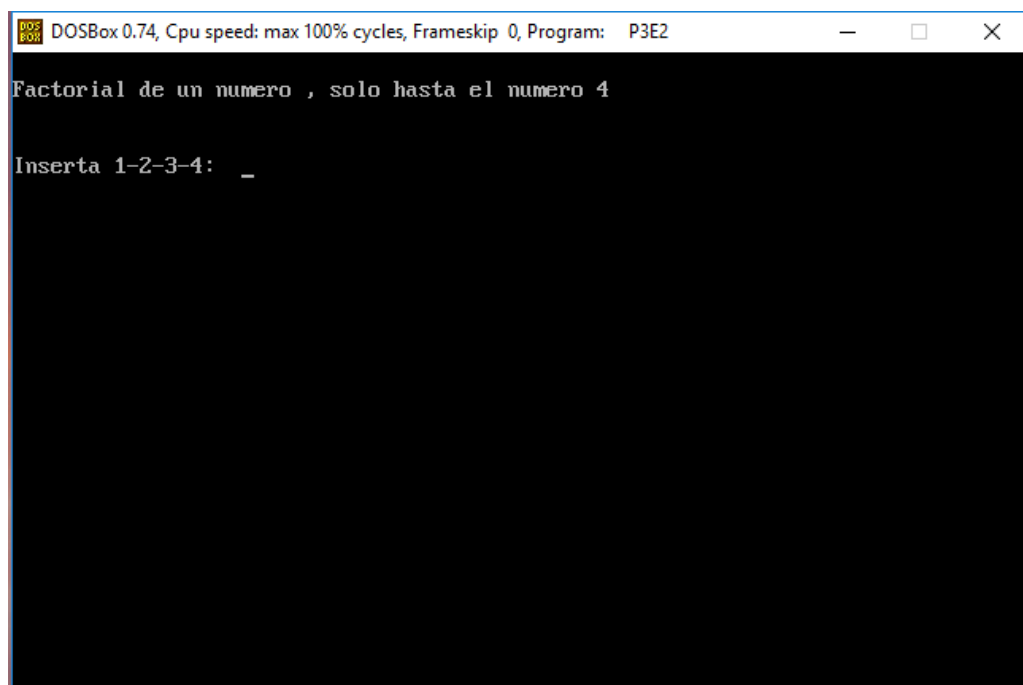


Ilustración 7 ejecución del programa

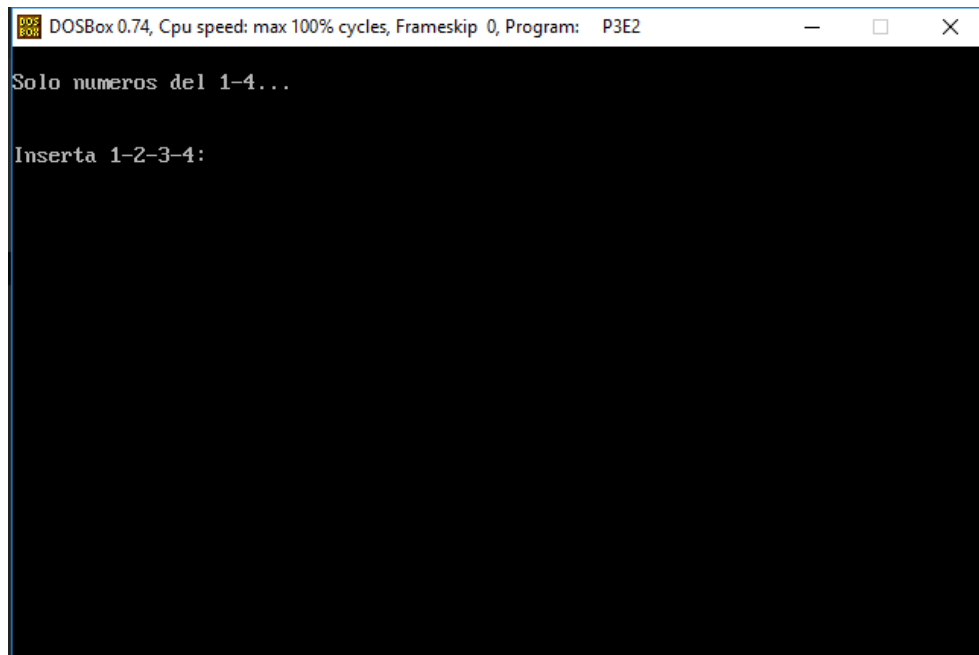


Ilustración 8 se inserta una tecla invalida

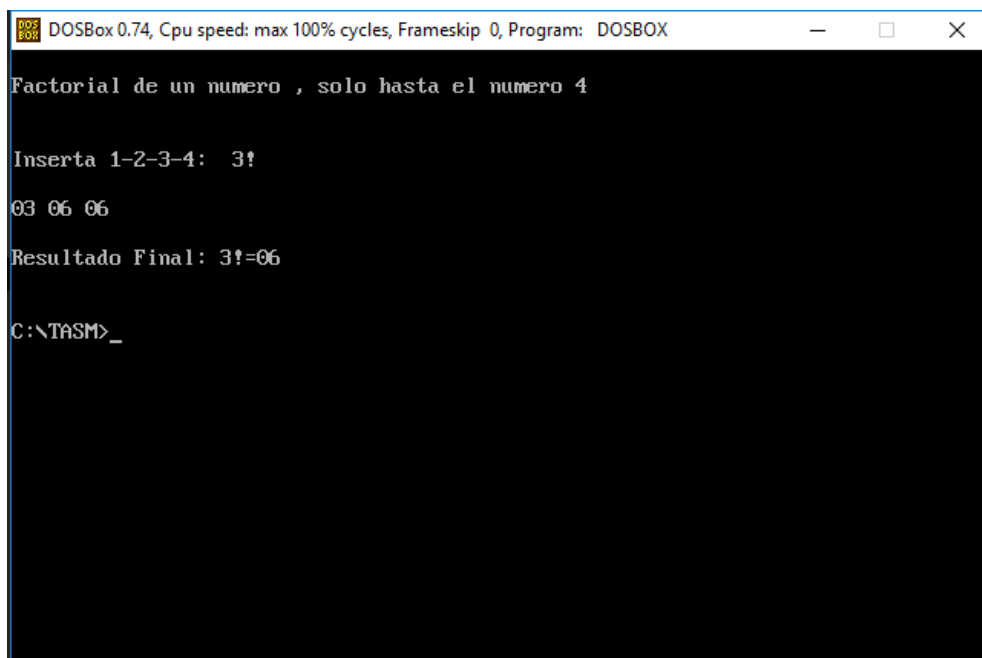


Ilustración 9 se inserta el numero 3

Nota₂: Como evidencia integre, el código fuente de cada uno de sus programas correctamente comentado, así como una impresión de pantalla de la ejecución del mismo, no olvide colocara pies de imagen debidamente referenciados.

Comentarios: Redacte comentarios acerca del aprendizaje adquirido una vez que desarrolle cada actividad de su práctica, así como las dudas que se generaron (extensión mínima 20 renglones).

El uso de operaciones con aritméticas en ensamblador es en una parte compleja, pues se tienen que tomar en cuenta que al insertar los valores desde el teclado se insertan en valores ASCII, es por eso por lo que debemos convertir a decimal, es decir solo le restamos los 30H si queremos operar con valores del cero al nueve, ya que los números se guardan del 30H al 39H, al valor que se guarda en el registro AL, pues solo podemos insertar un valor a la vez al igual que imprimir.

Si es necesario tener que trabajar con valores de más de dos dígitos, es necesario ir multiplicando el valor de entrada dependiendo de si es centena, decena, o unidad, en el caso de la unidad no tenemos que multiplicar para el valor, pues este solo es en unidades.

Para el uso de operadores aritméticos, como se está trabajando con valores ASCII, tenemos que realizar ajustes después de realizar operaciones, pues estas se realizan en números binarios por la CPU. Pero también puede realizar operaciones con cadenas de caracteres ASCII de decimales. Es por eso que se usan los ajustes ASCII, en donde hacen el desempaqueado de estos valores. Un problema que se tiene al usar estos ajustes es hacer operaciones que con las cuales se utilicen decenas en los valores. Pues estos ajustes solo funcionan con valores en unidades.

Al igual que de la forma en que se capturan los números de más de un dígito, como se van guardando en un registro en memoria, cada uno de los dígitos según corresponda, de esa misma manera se tienen que imprimir.

Podemos operar con valores de varios dígitos, el problema es la forma de ingresarlos al sistema o imprimirlos, pues solo podemos hacer la impresión de un carácter a la vez. Una vez que le hayamos movido una función cualquiera a un registro, podemos hacer las interrupciones cuantas veces queramos, y se podrán ejecutar las mismas acciones, siempre y cuando no le mandemos otra función.

Para la operación de división, en las referencias se encontró que el residuo se guardaba en la parte baja y el cociente en la parte alta del registro por el cual se multiplicaba, pero no se encontraba el valor del residuo, solo el del cociente.

Nota final: La copia total o parcial de alguna práctica previamente entregada no será recibida y afectará la calificación otorgada a la persona que entregó previamente. Sea cuidadoso con sus reportes y productos de las

actividades realizadas, evite pasarlos a sus compañeros. Si se basan en algún ejemplo de libros o de internet es preciso referenciarlo y hacer énfasis en las mejoras que usted género a dicho trabajo.

