



TECNOLÓGICO
NACIONAL DE MÉXICO



TECNOLÓGICO NACIONAL DE MEXICO

INSTITUTO TECNOLÓGICO DE OAXACA

Ingeniería en sistemas computacionales

Examen u3: Implementación de macros.

MATERIA:

Lenguajes de interfaz

PROFESOR:

Espinosa Pérez Jacob

ALUMNA:

Ramírez López Lorena Judith

Macros implementados al proyecto calculadora:

Macro que limpia la pantalla.

```
limpant      MACRO
              mov AH, 0Fh
              int 10h
              mov AH, 0
              int 10h
            |
            endm

imprime      MACRO cadena
              lea DX, cadena
              mov AH, 09
              int 21h
            endm

scand        MACRO
              mov AH, 01
              int 21h
            endm

restR        MACRO
              sub AX, AX
              sub BX, BX
              sub CX, CX
              sub DX, DX
            endm
```

Macro que imprime una cadena de caracteres, para restar con acarreo y para multiplicar.

```
SUMAR        MACRO SNA,SNB
              sub SNA, 30h
              sub SNB, 30h
              MOV AL, SNA
              MOV AH, 00
              ADD AL, SNB
              ADD AL, carry
              AAM
              MOV carry, AH
              mov AH, 00
            ENDM

RESTAR       MACRO RNA,RNB
              MOV AL, RNA
              MOV AH, 00
              ADD AL, carry
              SUB AL, RNB
              AAS
              MOV carry, AH
              mov AH, 00
            ENDM

multiplicar  MACRO MNA,MNB
              mov ah,00
              mov Al, MNA
              sub AL, 30h
              mov CH, MNB
              sub CH, 30h
              mul CH
              aam
              mov carry,ah
            ENDM
```

Macro para acarreo, para suma de resultados, y para ajuste de código ASCII

```
ajustemul    MACRO    V1, V2
              add     al, V1
              add     v2, ah
              mov     ah, 00
              aam
              mov     v1, al
              add     v2, ah
ENDM

sumamulc     MACRO    SMCA, SMCB
              mov     ah, 0
              mov     al, SMCA
              add     al, SMCB
              aam
              mov     dl, ah
              mov     ah, 00
ENDM

enmascm      MACRO

              or      alto1A, 30h
              or      alto2A, 30h
              or      bajo1A, 30h
              or      bajo2A, 30h

              or      alto1B, 30h
              or      alto2B, 30h
              or      bajo1B, 30h
              or      bajo2B, 30h
ENDM
```

Macros para Decimales y para quitar un decimal

```
anddivi      MACRO

              and     dive1, 0Fh
              and     dive2, 0Fh
              and     divd1, 0Fh
              and     divd2, 0Fh
ENDM

moverV       MACRO    valor
              sub     AH, AH
              mov     AL, dive1
              mul     mil
              mov     valor, AX

              sub     AH, AH
              mov     AL, dive2
              mul     cien
              add     valor, AX

              sub     AH, AH
              mov     AL, divd1
              mul     diez
              add     valor, AX

              sub     AH, AH
              mov     AL, divd2
              add     valor, AX
ENDM
```

Macro para ingresar datos:

```
MACRO piddatoA restR
enteroA:
    scand
    cmp AL, 2Eh
    je decimal1A

    cmp AL, 2Dh
    je sigA
    cmp AL, 0Dh
    je unidA

    push AX
    inc CX
    jmp enteroA

decimal1A:
    scand

    cmp AL, 0Dh
    je unidA

    cmp DL, 01
    je decimal2A

    mov bajo1A, AL
    inc DL
    jmp decimal1A

decimal2A:
    mov bajo2A, AL
    jmp unidA

sigA:
    inc signoA
    jmp enteroA
```

Menú:

```

;Definicion de cadenas de caracteres
-----
menums j db 10, 10, 09, 09, 'CALCULADORA U3', 13, 10
db 10, 10, 09, 09, '1.- Suma'
db 10, 10, 09, 09, '2.- Resta'
db 10, 10, 09, 09, '3.- Division'
db 10, 10, 09, 09, '4.- Multiplicacion'
db 10, 10, 09, 09, '5.- Salir', 13, 10, '$'
pdatoA db 10, 10, 'introduzca su primero valor: $'
pdatoB db 10, 10, 'introduzca su segundo valor: $'
opmenu db 10, 10, 10, 09, 09, 'Seleccione una operacion: $'
retmenu db 10, 10, 10, 'Deseas continuar?' una operacion: $'
db 10, 10, 10, '1.- Si'
db 10, 10, 10, '2.- No'
db 10, 10, 09, 'Seleccione una Operacion: $'
erro db 10, 10, 10, 10, 09, 09, 'OPCION INCORRECTA, INTENTELO DE NUEVO$', 13, 10
sms j db 10, 10, 09, 09, 09, 09, 'SUMA$'
rms j db 10, 10, 09, 09, 09, 09, 'RESTA$'
dms j db 10, 10, 09, 09, 09, 09, 'DIVISION$'
mms j db 10, 10, 09, 09, 09, 09, 'MULTIPLICACION$'
resul db 10, 10, 10, 09, 09, 'El resultado es: $'
despe db 10, 10, 09, 'Fin del programa. Adios.$' $'
indeter db 'ERROR. DIVISION NO VALIDA$'
negativo db '$'
punto db '$'
.CODE
inicia: mov AX, @data ;Inicia programa
mov DS, AX ;datos --> AX
;AX --> DS

fin: call MENU ;Llama a la rutina MENU
;Inicia fin
mov AX, 4Ch ;Regresa control al DOS
int 21h ;interrupcion de DOS.

MENU proc ;incia procedimiento menu
limpant ;Limpia pantalla
imprime menums j ;Despliega el menu
imprime pdatoA ;Pide primer numero
piddatoA ;Macro guarda valores
imprime pdatoB ;Pide segundo numero
piddatoB ;Macro guarda valores

enmascm ;Enmascara A y B
imprime opmenu ;Pide la operacion a realizar
scand ;inicia macro scand
cmp AL, '1' ;Compara AL con '1'
je SUMA ;Salta a SUMA
cmp AL, '2' ;Compara AL con '2'
je RESTAe ;Salta a RESTA
cmp AL, '3' ;Compara AL con '3'
je DIVIe ;Salta a DIVI
cmp AL, '4' ;Compara AL con '4'
je MULTIe ;Salta a MULTI
cmp AL, '5' ;Compara AL con '5'
je SALIRe ;Salta a SALIR
cmp AL, 00 ;Compara AL con 0
jne ERRORe ;Salta a error

```

Suma:

```
SUMA      proc
limpant           ;Limpia pantalla
imprime smsj      ;Imprime "SUMA"
imprime resul     ;imprime resultado
cmp signoA, 01    ;Analiza si numA es negativo
je  SAneg         ;Si lo es, salta a SAneg
cmp signoB, 01    ;Analiza si numB es negativo
je  SBneg         ;Si lo es, salta a SBneg
jmp su           ;Salta a su

SAneg:         ;Inicia SAneg
cmp signoB, 01    ;Analiza si numB es negativo
je  SABneg       ;Si lo es, salta a SABneg
jmp RABneg       ;Salta a -xx.xx + xx.xx

SABneg:        ;Inicia SABneg
imprime negativo  ;Imprime "-"
jmp su          ;Salta a su

SBneg:         ;Inicia SBneg
jmp re         ;Salta a re

su:           ;Inicia su
sumar bajo2A, bajo2B ;Suma (bajo2A + bajo2B)
push AX        ;Resultado a pila
sumar bajo1A, bajo1B ;Suma (bajo1A + bajo1B)
push AX        ;Resultado a pila
sumar alto2A, alto2B ;Suma (alto2A + alto2B)
push AX        ;Resultado a pila
sumar alto1A, alto1B ;Suma (alto1A + alto1B)
push AX        ;Resultado a pila
sumar alto2A, alto2B ;Suma (alto2A + alto2B)
push AX        ;Resultado a pila
sumar alto1A, alto1B ;Suma (alto1A + alto1B)
push AX        ;Resultado a pila
mov CX, 02      ;Contador antes del punto

sufinp:        ;Inicia sufinp
pop DX          ;Extrae valor de pila
add DL, 30h     ;Ajusta DL
mov AH, 02      ;Funcion 02...
int 21h         ;Imprime valor
loop sufinp     ;Retorna a sufinp
imprime punto   ;Imprime punto XX.
mov CX, 02      ;Contador para decimales

sufip:         ;Inicia sufip
pop DX          ;Imprime decimales .XX
add DL, 30h     ;Ajusta DL
mov AH, 02      ;Funcion 02
int 21h         ;Imprime valor
loop sufip     ;Retorna a sufip

restV          ;Restituye variables
restR          ;Restituye registros

imprime retmenu ;Imprime retmenu
scand          ;Pide dato de opción
cmp AL, '1'    ;Compara AL con '1'
je  MENUe      ;Salta a MENU
cmp AL, '2'    ;Compara AL con '2'
je  SALIRe2    ;Salta a SALIR

menue:        jmp MENU
salire2:      jmp SALIR
SUMA      endp
```

Resta:

| | |
|--|---|
| <pre> RESTA proc near lim pant imprime rmsj imprime resul cmp signoA, 01 je RAneg cmp signoB, 01 je ree jmp RBneg ree: jmp re RAneg: cmp signoB, 01 je SABnege jmp RABneg SABnege: jmp SABneg RBneg: jmp su RABneg: mov DL, alto1B cmp alto1A, DL ja mayorA jb mayorBe mov D1, alto2B cmp alto2A, DL ja mayorA jb mayorB mov DL, bajo1B cmp bajo1A, DL ja mayorA </pre> | <pre> ;Limpia pantalla. ;Imprime mensaje 'RESTA'. ;imprime resul. ;Analiza si numA es negativo. ;Si lo es, salta a RAneg. ;Analiza si numB es negativo. ;Si lo es, salta a RBneg. ;salta a resta xx.xx - xx.xx. ;Inicia RAneg ;Analiza si numB es negativo. ;Si lo es, salta a - xx.xx - (- xx.xx). ;Salta - xx.xx - xx.xx. ;Inicia RBneg ;Salta xx.xx - (- xx.xx). ;Comienza operacion - xx.xx - (- xx.xx). ;Asigna alto1b a DL. ;Compara alto1A con alto1B. ;Salta a resta mayorA. ;Salta a resta mayorB. ;Asigna alto2B a DL. ;Compara alto1A con alto1B. ;Salta a resta mayorA. ;Salta a resta mayorB. ;Asigna bajo1B a DL. ;Compara bajo1A con bajo1B. ;Salta a resta mayorA. </pre> |
|--|---|

División:

| | |
|--|--|
| <pre> DIVI proc near lim pant imprime dmsj imprime resul mov AL, alto1B mov dive1, AL mov AL, alto2B mov dive2, AL mov AL, bajo1B mov divd1, AL mov AL, bajo2B mov divd2, AL anddivi moverV divisor cmp divisor, 0 je diviindet jmp diviinic diviindet: imprime indeter jmp divifin diviinic: cmp signoA, 01 je DANeg cmp signoB, 01 je Dresulneg jmp indiv </pre> | <pre> ;Inicia procedimiento DIVI ;Limpia pantalla ;Imprime mensaje 'DIVISION' ;imprime resul ;< Divisor -> Variables auxs... ; ...Para quitar'. ;Convierte a decimal ;Si divisor = 0,... ;... salta a diviindet ;Salta inicio de la division ;Imprime mensaje de error ;Salta fin de la division ;Inicia division ;Verifica si numA es negativo ;Si lo es, salta a DANeg ;Verifica si numB es negativo ;Si lo es, salta a Dresulneg ;Salta a division </pre> |
|--|--|

Multiplicación:

```

MULTI      proc near
            lim pant          ;Limpia pantalla
            imprime mmsj      ;Imprime mensaje 'MULTIPLICACION'
            imprime resul     ;imprime resul

            restR
            enmascm

            cmp signoA, 01     ;Verifica si numA es negativo
            je  MAneg          ;Si lo es, salta a MAneg
            cmp signoB, 01     ;Verifica si numB es negativo
            je  Mresulneg      ;Si lo es, salta a Mresulneg

            jmp mu             ;Salta a multiplicacion

MAneg:
            cmp signoB, 01     ;Verifica si numB es negativo
            je  mu             ;Si lo es, salta a multiplicacion

            jmp Mresulneg      ;Salta a imprimir signo '-'

Mresulneg:
            imprime negativo   ;imprime '-'
            jmp mu             ; salta a mu (etiqueta multiplicacion)

mu:
            multiplicar bajo2a,bajo2b
            mov  e, al
            add  d, ah
            multiplicar bajo1a,bajo2b
            ajustemul d, c
            multiplicar alto2a,bajo2b
            ajustemul c, b

```

Opción Salir:

```

; procedimiento SALIR

SALIR      proc near
            lim pant          ;Limpia pantalla
            imprime despe     ;Imprime despedida
            scand              ;Hace una pausa

            jmp fin           ;Salta a fin

SALIR      endp

; procedimiento ERROR

ERROR      proc near
            lim pant          ;Limpia pantalla
            imprime erro      ;Avisa de un error
            scand              ;Hace una pausa
            restV              ;Restituye todas las variables
            jmp MENU          ;Salta a MENU

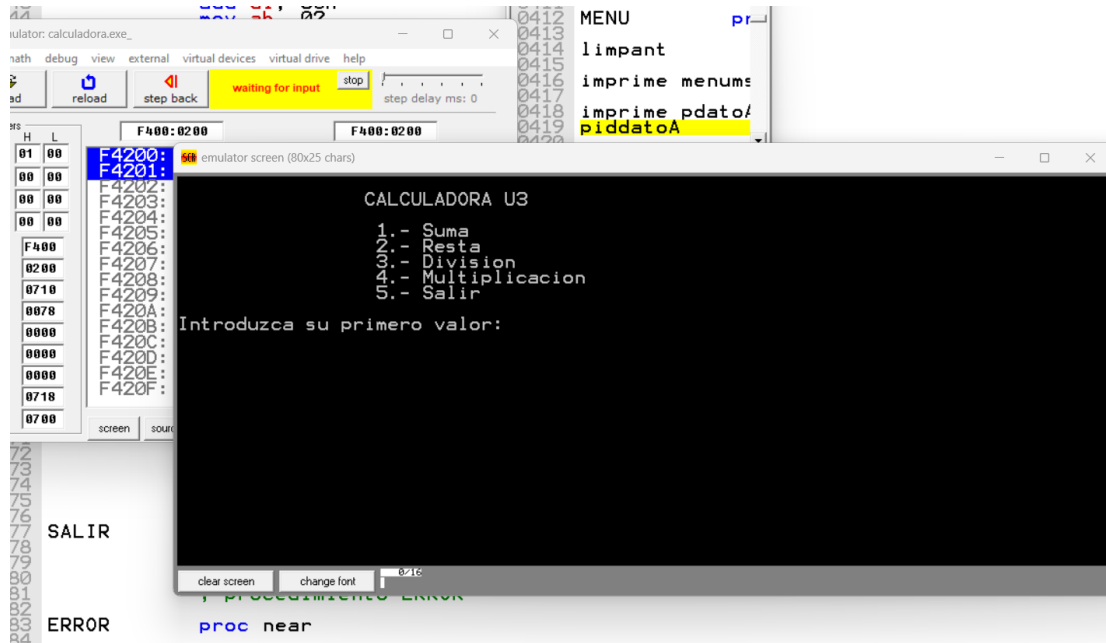
ERROR      endp

end inicia

```

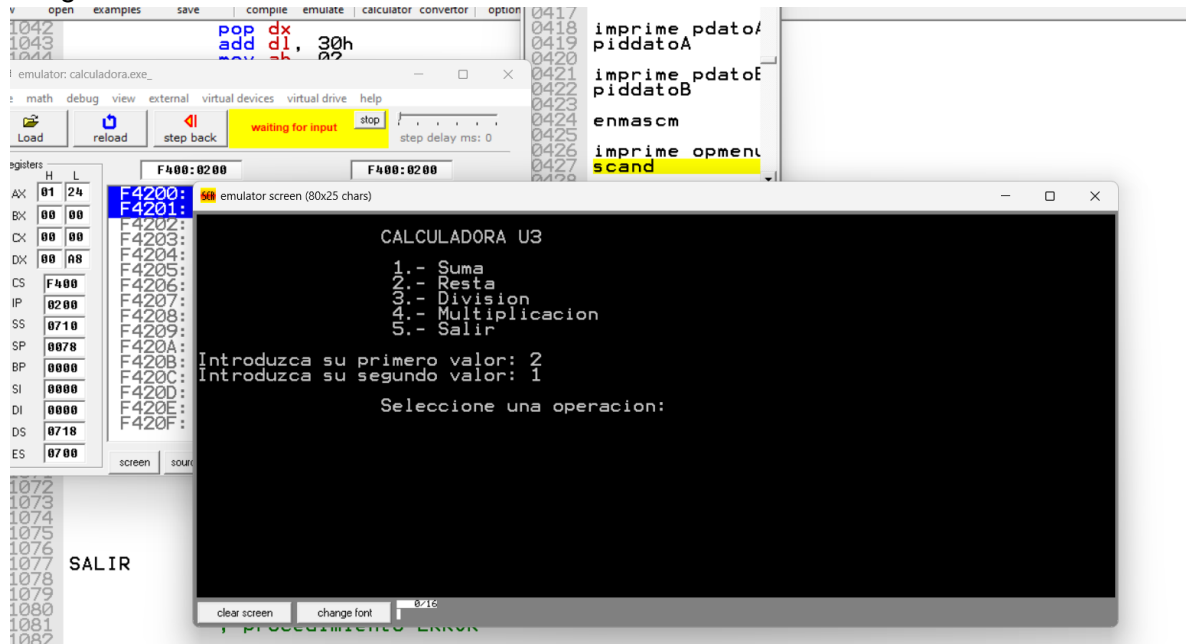

Pruebas de escritorio:

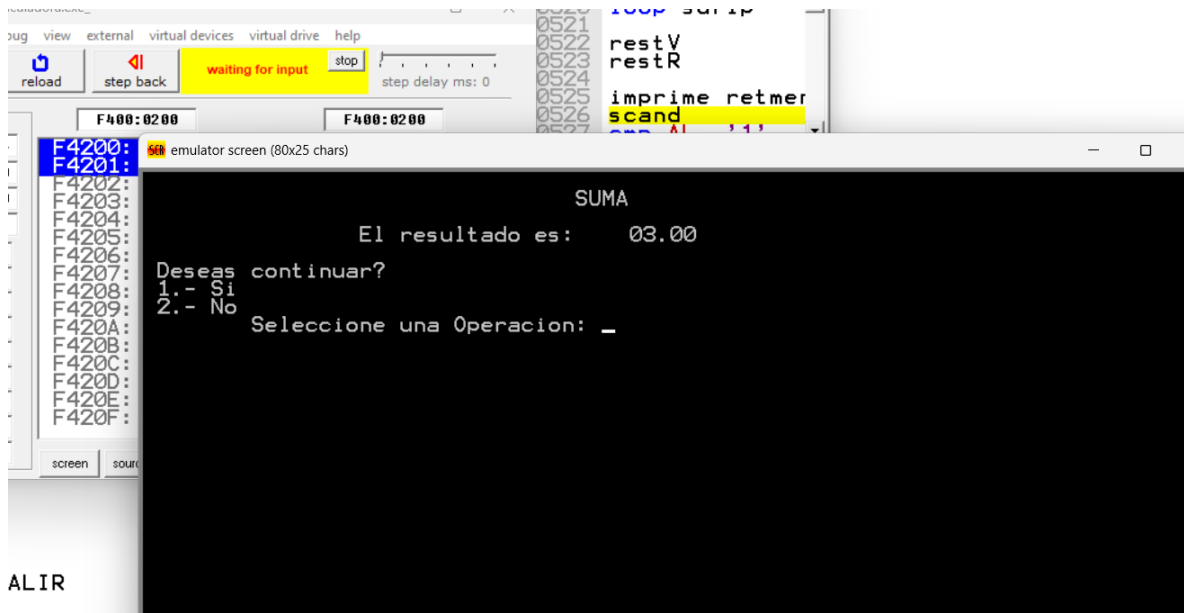
Menú:



Opción Suma:

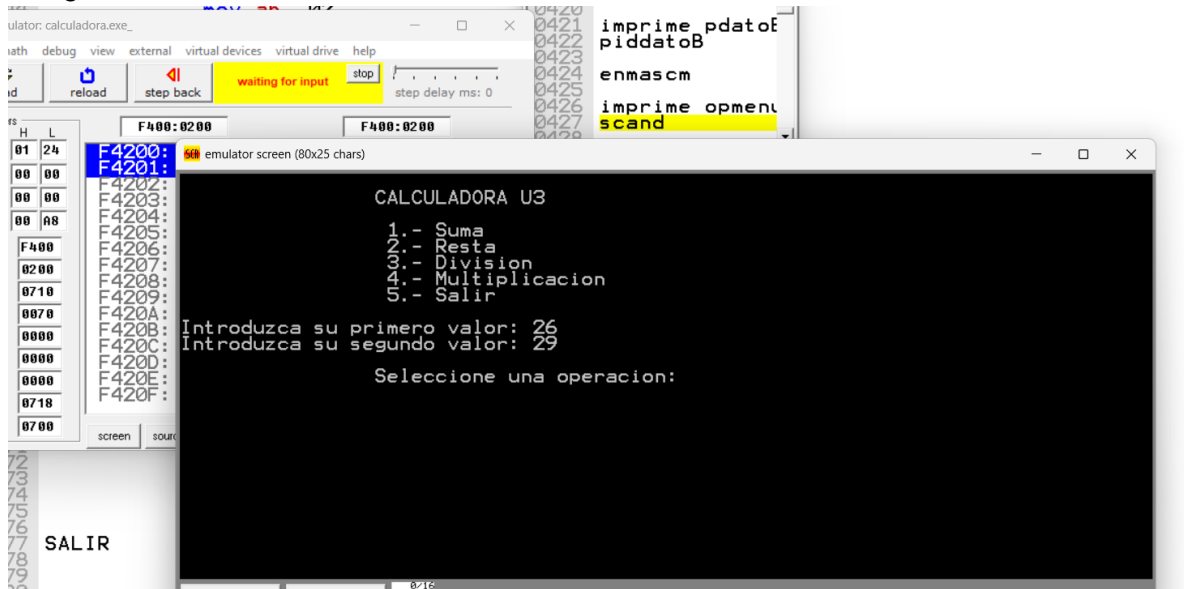
1 dígito:



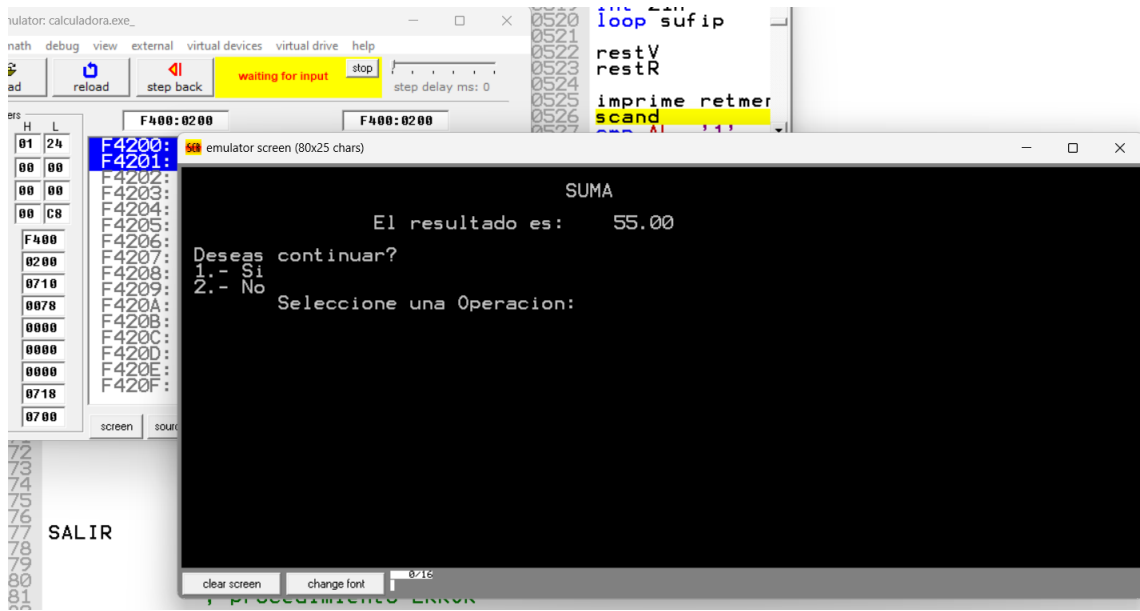


ALIR

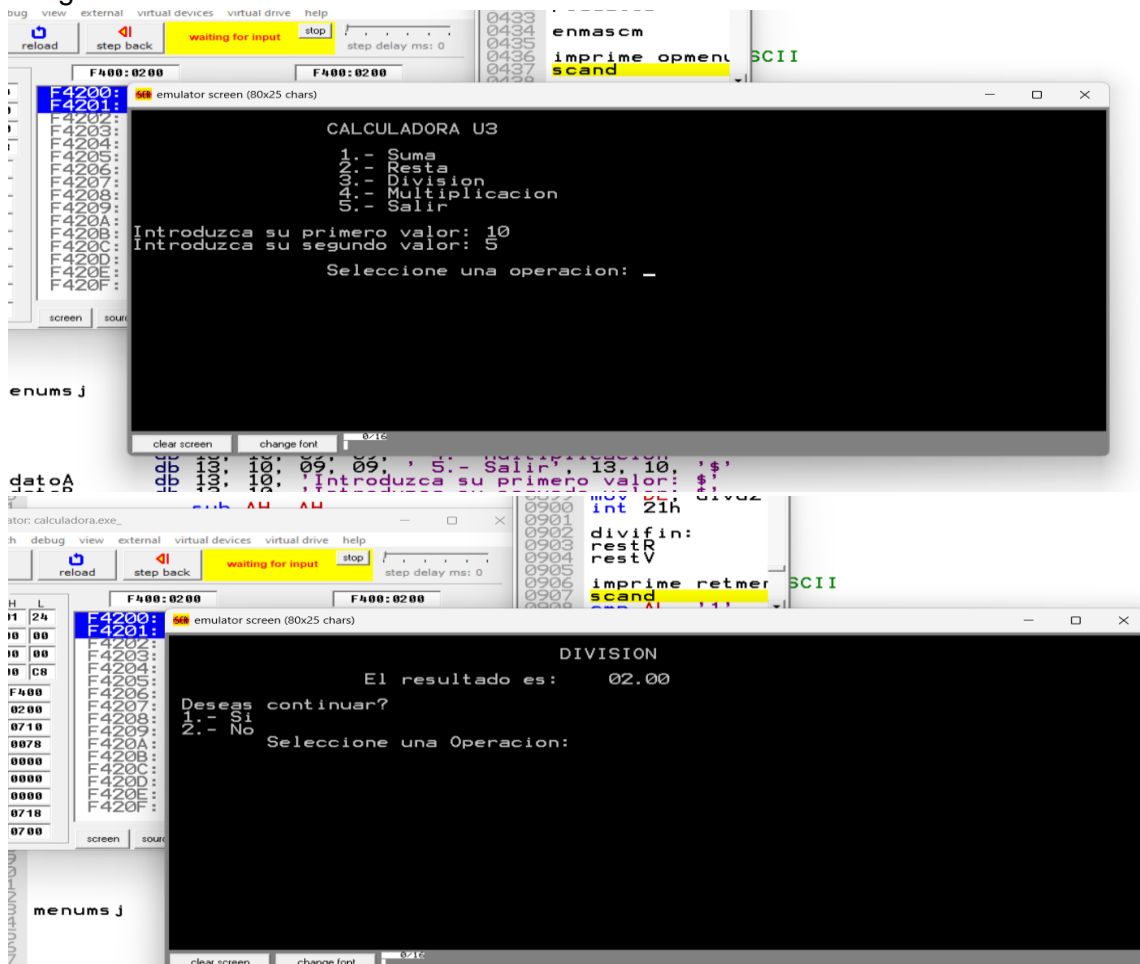
2 Dígitos:



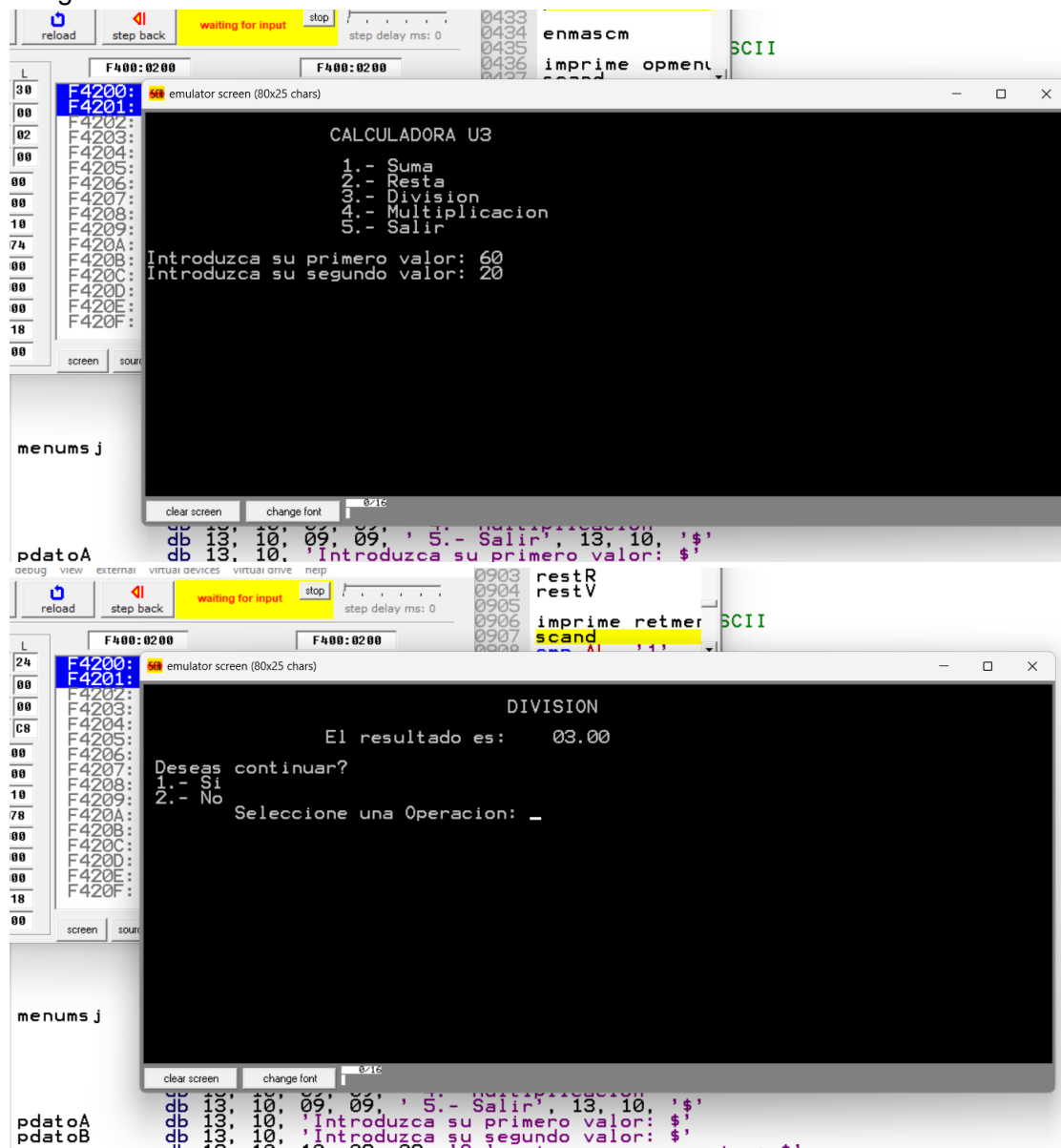
SALIR



División:
1 dígito:



2 dígitos:



1 dígito:

