**EXER25.ASM**

*; Filename: EXER25.ASM*

*; Programmer Name: JOHN KYLE A. VILLARAZO*

*; Date: 09/28/2024*

*; Description: This assembly language program will get 3 string inputs and display back the 3 strings on separate liens.*

.model small

.stack 100h

.data

    inputString1 *db* 50 dup ('$')

    inputString2 *db* 50 dup ('$')

    inputString3 *db* 50 dup ('$')

    promptString1 *db* 'Enter first string: $'

    promptString2 *db* 'Enter second string: $'

    promptString3 *db* 'Enter third string: $'

    outputString1 *db* 'You entered first string: $'

    outputString2 *db* 'You entered second string: $'

    outputString3 *db* 'You entered third string: $'

.code

Main Proc

*; Initialize data segment*

*mov* ax, @data

*mov* ds, ax

*; Prompt the user for input first string*

*lea* dx, promptString1

*mov* ah, 09h             *; DOS function to display a string*

*int* 21h

*; Read input first string*

*lea* dx, inputString1

*mov* ah, 0ah             *; DOS function to read a string*

*int* 21h

*; this is for displaying new line*

*mov* ah, 02h

*mov* cl, 0ah     *; 0ah is new line*

*mov* dl, cl

*int* 21h

*; Prompt the user for input second string*

*lea* dx, promptString2

*mov* ah, 09h

*int* 21h

*; Read input second string*

*lea* dx, inputString2

*mov* ah, 0ah             *; DOS function to read a string*

*int* 21h

*; this is for displaying new line*

*mov* ah, 02h

*mov* cl, 0ah     *; 0ah is new line*

*mov* dl, cl

*int* 21h

*; Prompt the user for input third string*

*lea* dx, promptString3

*mov* ah, 09h

*int* 21h

*; Read input third string*

*lea* dx, inputString3

*mov* ah, 0ah             *; DOS function to read a string*

*int* 21h

*; this is for displaying new line*

*mov* ah, 02h

*mov* cl, 0ah     *; 0ah is new line*

*mov* dl, cl

*int* 21h

*; Display the output message for first string*

*lea* dx, outputString1

*mov* ah, 09h             *; DOS function to display a string*

*int* 21h

*; Display the entered first string*

*lea* dx, inputString1 + 2    *; Skip the first two bytes*

*mov* ah, 09h             *; DOS function to display a string*

*int* 21h

*; this is for displaying new line*

*mov* ah, 02h

*mov* cl, 0ah     *; 0ah is new line*

*mov* dl, cl

*int* 21h

*; Display the output message for second string*

*lea* dx, outputString2

*mov* ah, 09h *; DOS function to display a string*

*int* 21h

*; Display the entered second string*

*lea* dx, inputString2 + 2    *; Skip the first two bytes*

*mov* ah, 09h             *; DOS function to display a string*

*int* 21h

*; this is for displaying new line*

*mov* ah, 02h

*mov* cl, 0ah     *; 0ah is new line*

*mov* dl, cl

*int* 21h

*; Display the output message for third string*

*lea* dx, outputString3

*mov* ah, 09h *; DOS function to display a string*

*int* 21h

*; Display the entered third string*

*lea* dx, inputString3 + 2    *; Skip the first two bytes*

*mov* ah, 09h             *; DOS function to display a string*

*int* 21h

*; this is for displaying new line*

*mov* ah, 02h

*mov* cl, 0ah     *; 0ah is new line*

*mov* dl, cl

*int* 21h

*mov* ax, 4C00h

*int* 21h

Main endp

End Main

**EXER26.ASM**

*; Filename: EXER26.ASM*

*; Programmer Name: JOHN KYLE A. VILLARAZO*

*; Date: 09/28/2024*

.model small

.stack 200h

.data

    num1    *dw* ?

    num2    *dw* ?

    sum     *dw* ?

    firstMsg    *db* 0dh, 0ah, 'Enter first integer: $'

    secondMsg   *db* 0dh, 0ah, 'Enter second integer: $'

    sumMsg      *db* 0dh, 0ah, 'Sum: $'

.code

Start:

*mov* ax, @data

*mov* ds, ax

*lea* dx, firstMsg

*mov* ah, 09h

*int* 21h

*mov* ah, 01h

*int* 21h

*sub* al, '0'

*mov* ah, 0

*mov* num1, ax

*lea* dx, secondMsg

*mov* ah, 09h

*int* 21h

*mov* ah, 01h

*int* 21h

*sub* al, '0'

*mov* ah, 0

*mov* num2, ax

*mov* dx, num1

*add* dx, num2

*mov* sum, dx

*lea* dx, sumMsg

*mov* ah, 09h

*int* 21h

*mov* ax, sum

*call* Convert

*mov* ax, 4C00h

*int* 21h

Convert Proc

*push* ax

*push* bx

*push* cx

*push* dx

*mov* cx, 0

*mov* bx, 10

    converter\_loop1:

*xor* dx, dx

*div* bx

*push* dx

*inc* cx

*cmp* ax, 0

*jne* converter\_loop1

    converter\_loop2:

*pop* dx

*add* dl, '0'

*mov* ah, 02h

*int* 21h

*dec* cx

*cmp* cx, 0

*jne* converter\_loop2

*pop* dx

*pop* cx

*pop* bx

*pop* ax

*ret*

Convert endp

end Start

**EXER27.ASM**

*; Filename: EXER27.ASM*

*; Programmer Name: JOHN KYLE A. VILLARAZO*

*; Date: 09/28/2024*

.model small

.stack 200h

.data

    num1    *dw* ?

    num2    *dw* ?

    diff    *dw* ?

    firstMsg    *db* 0dh, 0ah, 'Enter first integer: $'

    secondMsg   *db* 0dh, 0ah, 'Enter second integer: $'

    differenceMsg      *db* 0dh, 0ah, 'Difference: $'

.code

Start:

*mov* ax, @data

*mov* ds, ax

*lea* dx, firstMsg

*mov* ah, 09h

*int* 21h

*mov* ah, 01h

*int* 21h

*sub* al, '0'

*mov* ah, 0

*mov* num1, ax

*lea* dx, secondMsg

*mov* ah, 09h

*int* 21h

*mov* ah, 01h

*int* 21h

*sub* al, '0'

*mov* ah, 0

*mov* num2, ax

*mov* dx, num1

*sub* dx, num2

*mov* diff, dx

*lea* dx, differenceMsg

*mov* ah, 09h

*int* 21h

*mov* ax, diff

*call* Convert

*mov* ax, 4C00h

*int* 21h

Convert Proc

*push* ax

*push* bx

*push* cx

*push* dx

*mov* cx, 0

*mov* bx, 10

    converter\_loop1:

*xor* dx, dx

*div* bx

*push* dx

*inc* cx

*cmp* ax, 0

*jne* converter\_loop1

    converter\_loop2:

*pop* dx

*add* dl, '0'

*mov* ah, 02h

*int* 21h

*dec* cx

*cmp* cx, 0

*jne* converter\_loop2

*pop* dx

*pop* cx

*pop* bx

*pop* ax

*ret*

Convert endp

end Start

**EXER28.ASM**

*; Filename: EXER28.ASM*

*; Programmer Name: JOHN KYLE A. VILLARAZO*

*; Date: 09/28/2024*

.model small

.stack 200h

.data

    num1    *dw* ?

    num2    *dw* ?

    prod     *dw* ?

    firstMsg    *db* 0dh, 0ah, 'Enter first integer: $'

    secondMsg   *db* 0dh, 0ah, 'Enter second integer: $'

    productMsg      *db* 0dh, 0ah, 'Product: $'

.code

Start:

*mov* ax, @data

*mov* ds, ax

*lea* dx, firstMsg

*mov* ah, 09h

*int* 21h

*mov* ah, 01h

*int* 21h

*sub* al, '0'

*mov* ah, 0

*mov* num1, ax

*lea* dx, secondMsg

*mov* ah, 09h

*int* 21h

*mov* ah, 01h

*int* 21h

*sub* al, '0'

*mov* ah, 0

*mov* num2, ax

*mov* ax, num1

*mov* bx, num2

*mul* bx

*mov* prod, ax

*lea* dx, productMsg

*mov* ah, 09h

*int* 21h

*mov* ax, prod

*call* Convert

*mov* ax, 4C00h

*int* 21h

Convert Proc

*push* ax

*push* bx

*push* cx

*push* dx

*mov* cx, 0

*mov* bx, 10

    converter\_loop1:

*xor* dx, dx

*div* bx

*push* dx

*inc* cx

*cmp* ax, 0

*jne* converter\_loop1

    converter\_loop2:

*pop* dx

*add* dl, '0'

*mov* ah, 02h

*int* 21h

*dec* cx

*cmp* cx, 0

*jne* converter\_loop2

*pop* dx

*pop* cx

*pop* bx

*pop* ax

*ret*

Convert endp

end Start

**EXER29.ASM**

*; Filename: EXER29.ASM*

*; Programmer Name: JOHN KYLE A. VILLARAZO*

*; Date: 09/28/2024*

.model small

.stack 200h

.data

    num1    *dw* ?

    num2    *dw* ?

    quot    *dw* ?

    firstMsg    *db* 0dh, 0ah, 'Enter first integer: $'

    secondMsg   *db* 0dh, 0ah, 'Enter second integer: $'

    quotientMsg  *db* 0dh, 0ah, 'Quotient: $'

.code

Start:

*mov* ax, @data

*mov* ds, ax

*lea* dx, firstMsg

*mov* ah, 09h

*int* 21h

*mov* ah, 01h

*int* 21h

*sub* al, '0'

*mov* ah, 0

*mov* num1, ax

*lea* dx, secondMsg

*mov* ah, 09h

*int* 21h

*mov* ah, 01h

*int* 21h

*sub* al, '0'

*mov* ah, 0

*mov* num2, ax

*mov* ax, num1

*mov* bx, num2

*xor* ah, ah

*div* bl

*mov* quot, ax

*lea* dx, quotientMsg

*mov* ah, 09h

*int* 21h

*mov* ax, quot

*call* Convert

*mov* ax, 4C00h

*int* 21h

Convert Proc

*push* ax

*push* bx

*push* cx

*push* dx

*mov* cx, 0

*mov* bx, 10

    converter\_loop1:

*xor* dx, dx

*div* bx

*push* dx

*inc* cx

*cmp* ax, 0

*jne* converter\_loop1

    converter\_loop2:

*pop* dx

*add* dl, '0'

*mov* ah, 02h

*int* 21h

*dec* cx

*cmp* cx, 0

*jne* converter\_loop2

*pop* dx

*pop* cx

*pop* bx

*pop* ax

*ret*

Convert endp

end Start

**EXER30.ASM**

*; Filename: EXER30.ASM*

*; Programmer Name: JOHN KYLE A. VILLARAZO*

*; Date: 09/28/2024*

.model small

.stack 200h

.data

    num1    *db* ?

    num2    *db* ?

    firstMsg    *db* 0dh, 0ah, 'Enter first integer: $'

    secondMsg   *db* 0dh, 0ah, 'Enter second integer: $'

    bigMsg      *db* 0dh, 0ah, 'The bigger integer is: $'

    smallMsg    *db* 0dh, 0ah, 'The smaller integer is: $'

    equalMsg    *db* 0dh, 0ah, 'Both are equal$'

.code

Start:

*mov* ax, @data

*mov* ds, ax

*lea* dx, firstMsg

*mov* ah, 09h

*int* 21h

*mov* ah, 01h

*int* 21h

*sub* al, '0'    *; Convert from ASCII*

*mov* num1, al

*lea* dx, secondMsg

*mov* ah, 09h

*int* 21h

*mov* ah, 01h

*int* 21h

*sub* al, '0'    *; Convert from ASCII*

*mov* num2, al

*mov* al, num1

*cmp* al, num2

*jg* Bigger

*jl* Smaller

*jmp* Equal

Bigger:

*mov* dx, offset bigMsg

*mov* ah, 09h

*int* 21h

*mov* al, num1

*add* al, '0'

*mov* dl, al

*mov* ah, 02h

*int* 21h

*mov* dx, offset smallMsg

*mov* ah, 09h

*int* 21h

*mov* al, num2

*add* al, '0'

*mov* dl, al

*mov* ah, 02h

*int* 21h

*jmp* Exit

Smaller:

*mov* dx, offset bigMsg

*mov* ah, 09h

*int* 21h

*mov* al, num2

*add* al, '0'

*mov* dl, al

*mov* ah, 02h

*int* 21h

*mov* dx, offset smallMsg

*mov* ah, 09h

*int* 21h

*mov* al, num1

*add* al, '0'

*mov* dl, al

*mov* ah, 02h

*int* 21h

*jmp* Exit

Equal:

*mov* dx, offset equalMsg

*mov* ah, 09h

*int* 21h

*jmp* Exit

Exit:

*mov* ah, 02h

*mov* cl, 0ah

*mov* dl, cl

*int* 21h

*mov* ax, 4C00h

*int* 21h

end Start

**EXER31.ASM**

*; Filename: EXER31.ASM*

*; Programmer Name: JOHN KYLE A. VILLARAZO*

*; Date: 09/28/2024*

.model small

.stack 200h

.data

    num         *db* 0

    message     *db* 0dh, 0ah, 'Enter an integer: $'

    newLine     *db* 0dh, 0ah, '$'

.code

Start:

*mov* ax, @data

*mov* ds, ax

*lea* dx, message

*mov* ah, 09h

*int* 21h

*mov* ah, 01h

*int* 21h

*sub* al, '0'

*mov* num, al

*mov* cl, 1

*lea* dx, newLine

*mov* ah, 09h

*int* 21h

LoopInteger:

*cmp* cl, num

*jg* Exit

*mov* al, cl

*add* al, '0'

*mov* dl, al

*mov* ah, 02h

*int* 21h

*lea* dx, newline

*mov* ah, 09h

*int* 21h

*inc* cl

*jmp* LoopInteger

Exit:

*mov* ax, 4C00h

*int* 21h

end Start

**EXER32.ASM**

*; Filename: EXER32.ASM*

*; CS243 Lab Hands-on Exam No. 2*

*; First Semester SY 2024-2025*

*; Student Name: JOHN KYLE A. VILLARAZO*

*; Date Finished: 09/28/2024*

.model small

.stack 200h

.data

    choice *dw* ?

    num1 *dw* ?

    num2 *dw* ?

    sum *dw* ?

    difference *dw* ?

    product *dw* ?

    quotient *dw* ?

    calculator      *db* 0dh, 0ah, 'THE CALCULATOR', 0dh, 0ah

*db* 'Created by: JOHN KYLE A. VILLARAZO', 0dh, 0ah

*db* 'Date: 09/28/2024', 0dh, 0ah, 0dh, 0ah

*db* 'Main Menu', 0dh, 0ah

*db* 'a - Addition', 0dh, 0ah

*db* 's - Subtraction', 0dh, 0ah

*db* 'm - Multiplication', 0dh, 0ah

*db* 'd - Division', 0dh, 0ah

*db* 'e - Exit', 0dh, 0ah

*db* 'Enter your choice: $'

    addition        *db* 0dh, 0ah, 0dh, 0ah, 'Addition', 0dh, 0ah

*db* 'Enter first addend: $'

    addition2       *db* 0dh, 0ah, 'Enter second addend: $'

    addend1         *db* 0dh, 0ah, 'First addend is: $'

    addend2         *db* 0dh, 0ah, 'Second addend is: $'

    subtraction     *db* 0dh, 0ah, 0dh, 0ah, 'Subtraction', 0dh, 0ah

*db* 'Enter minuend: $'

    subtraction2    *db* 0dh, 0ah, 'Enter subtrahend: $'

    minuend1        *db* 0dh, 0ah, 'Minuend is: $'

    minuend2        *db* 0dh, 0ah, 'Subtrahend is: $'

    multiplication  *db* 0dh, 0ah, 0dh, 0ah, 'Multiplication', 0dh, 0ah

*db* 'Enter multiplicand: $'

    multiplication2 *db* 0dh, 0ah, 'Enter multiplier: $'

    multiplicand1   *db* 0dh, 0ah, 'Multiplicand is: $'

    multiplicand2   *db* 0dh, 0ah, 'Multiplier is: $'

    division        *db* 0dh, 0ah, 0dh, 0ah, 'Division', 0dh, 0ah

*db* 'Enter dividend: $'

    division2       *db* 0dh, 0ah, 'Enter divisor: $'

    dividend1       *db* 0dh, 0ah, 'Dividend is: $'

    dividend2       *db* 0dh, 0ah, 'Divisor is: $'

    endProgram      *db* 0dh, 0ah, 0dh, 0ah, 'Exit Program$'

    invalid         *db* 0dh, 0ah, 0dh, 0ah, 'INVALID CHOICE!$'

    thankYou        *db* 0dh, 0ah, 0dh, 0ah, 'Thank you.', 0dh, 0ah, '$'

    sumMessage      *db* 0dh, 0ah, 'Sum: $'

    prodMessage     *db* 0dh, 0ah, 'Product: $'

    diffMessage     *db* 0dh, 0ah, 'Difference: $'

    quotMessage     *db* 0dh, 0ah, 'Quotient: $'

    continue        *db* 0dh, 0ah, 0dh, 0ah, 'Press enter to continue.$'

.code

ProgramStart:

*mov* ax, @data

*mov* ds, ax

*mov* ax, 3

*int* 10h

*; Background Colors*

*mov* ah, 06h

*xor* al, al

*xor* cx, cx

*mov* ch, 6

*mov* cl, 0

*mov* dh, 6

*mov* dl, 11

*mov* bh, 68h

*int* 10h

*mov* ah, 06h

*mov* ch, 7

*mov* cl, 0

*mov* dh, 7

*mov* dl, 14

*mov* bh, 28h

*int* 10h

*mov* ah, 06h

*mov* ch, 8

*mov* cl, 0

*mov* dh, 8

*mov* dl, 17

*mov* bh, 18h

*int* 10h

*mov* ah, 06h

*mov* ch, 9

*mov* cl, 0

*mov* dh, 9

*mov* dl, 11

*mov* bh, 58h

*int* 10h

*mov* ah, 06h

*mov* ch, 10

*mov* cl, 0

*mov* dh, 10

*mov* dl, 7

*mov* bh, 78h

*int* 10h

*lea* dx, calculator

*mov* ah, 09h

*int* 21h

*mov* ah, 01h

*int* 21h

*mov* choice, ax

*cmp* al, 'a'

*je* caseAdd

*cmp* al, 's'

*je* caseSubtract

*cmp* al, 'm'

*je* caseMultiply

*cmp* al, 'd'

*je* caseDivide

*cmp* al, 'e'

*je* caseExit

*jmp* caseInvalid

caseAdd:

*call* AdditionProc

*call* DisplayContinue

*jmp* ProgramStart

caseSubtract:

*call* SubtractionProc

*call* DisplayContinue

*jmp* ProgramStart

caseMultiply:

*call* MultiplicationProc

*call* DisplayContinue

*jmp* ProgramStart

caseDivide:

*call* DivisionProc

*call* DisplayContinue

*jmp* ProgramStart

caseExit:

*call* Exit

*call* DisplayContinue

*jmp* ExitProgram

caseInvalid:

*call* InvalidInputProc

*jmp* ProgramStart

AdditionProc Proc

*mov* ah, 06h

*xor* al, al

*xor* cx, cx

*mov* ch, 13

*mov* cl, 0

*mov* dh, 13

*mov* dl, 7

*mov* bh, 68h

*int* 10h

*lea* dx, addition

*mov* ah, 09h

*int* 21h

*call* Input

*mov* num1, ax

*lea* dx, addition2

*mov* ah, 09h

*int* 21h

*call* Input

*mov* num2, ax

*lea* dx, addend1

*mov* ah, 09h

*int* 21h

*mov* ax, num1

*call* DisplayInput

*lea* dx, addend2

*mov* ah, 09h

*int* 21h

*mov* ax, num2

*call* DisplayInput

*lea* dx, sumMessage

*mov* ah, 09h

*int* 21h

*mov* dx, num1

*add* dx, num2

*mov* sum, dx

*mov* ax, sum

*call* Convert

*ret*

AdditionProc endp

SubtractionProc Proc

*mov* ah, 06h

*xor* al, al

*xor* cx, cx

*mov* ch, 13

*mov* cl, 0

*mov* dh, 13

*mov* dl, 10

*mov* bh, 28h

*int* 10h

*lea* dx, subtraction

*mov* ah, 09h

*int* 21h

*call* Input

*mov* num1, ax

*lea* dx, subtraction2

*mov* ah, 09h

*int* 21h

*call* Input

*mov* num2, ax

*lea* dx, minuend1

*mov* ah, 09h

*int* 21h

*mov* ax, num1

*call* DisplayInput

*lea* dx, minuend2

*mov* ah, 09h

*int* 21h

*mov* ax, num2

*call* DisplayInput

*lea* dx, diffMessage

*mov* ah, 09h

*int* 21h

*mov* dx, num1

*sub* dx, num2

*mov* difference, dx

*mov* ax, difference

*call* Convert

*ret*

SubtractionProc endp

MultiplicationProc Proc

*mov* ah, 06h

*xor* al, al

*xor* cx, cx

*mov* ch, 13

*mov* cl, 0

*mov* dh, 13

*mov* dl, 13

*mov* bh, 18h

*int* 10h

*lea* dx, multiplication

*mov* ah, 09h

*int* 21h

*call* Input

*mov* num1, ax

*lea* dx, multiplication2

*mov* ah, 09h

*int* 21h

*call* Input

*mov* num2, ax

*lea* dx, multiplicand1

*mov* ah, 09h

*int* 21h

*mov* ax, num1

*call* DisplayInput

*lea* dx, multiplicand2

*mov* ah, 09h

*int* 21h

*mov* ax, num2

*call* DisplayInput

*lea* dx, prodMessage

*mov* ah, 09h

*int* 21h

*mov* ax, num1

*mov* bx, num2

*mul* bx

*mov* product, ax

*mov* ax, product

*call* Convert

*ret*

MultiplicationProc endp

DivisionProc Proc

*mov* ah, 06h

*xor* al, al

*xor* cx, cx

*mov* ch, 13

*mov* cl, 0

*mov* dh, 13

*mov* dl, 7

*mov* bh, 58h

*int* 10h

*lea* dx, division

*mov* ah, 09h

*int* 21h

*call* Input

*mov* num1, ax

*lea* dx, division2

*mov* ah, 09h

*int* 21h

*call* Input

*mov* num2, ax

*lea* dx, dividend1

*mov* ah, 09h

*int* 21h

*mov* ax, num1

*call* DisplayInput

*lea* dx, dividend2

*mov* ah, 09h

*int* 21h

*mov* ax, num2

*call* DisplayInput

*lea* dx, quotMessage

*mov* ah, 09h

*int* 21h

*mov* ax, num1

*mov* bx, num2

*xor* ah, ah

*div* bl

*mov* quotient, ax

*mov* ax, quotient

*call* Convert

*ret*

DivisionProc endp

Convert Proc

*push* ax

*push* bx

*push* cx

*push* dx

*mov* cx, 0

*mov* bx, 10

    converter\_loop1:

*xor* dx, dx

*div* bx

*push* dx

*inc* cx

*cmp* ax, 0

*jne* converter\_loop1

    converter\_loop2:

*pop* dx

*add* dl, '0'

*mov* ah, 02h

*int* 21h

*dec* cx

*cmp* cx, 0

*jne* converter\_loop2

*pop* dx

*pop* cx

*pop* bx

*pop* ax

*ret*

Convert endp

Input Proc

*mov* ah, 01h

*int* 21h

*sub* al, '0'

*mov* ah, 0

*ret*

Input endp

DisplayInput Proc

*add* al, '0'

*mov* dl, al

*mov* ah, 02h

*int* 21h

*ret*

DisplayInput endp

Exit Proc

*mov* ah, 06h

*xor* al, al

*xor* cx, cx

*mov* ch, 13

*mov* cl, 0

*mov* dh, 13

*mov* dl, 11

*mov* bh, 78h

*int* 10h

*lea* dx, endProgram

*mov* ah, 09h

*int* 21h

*ret*

Exit endp

InvalidInputProc Proc

*mov* ah, 06h

*xor* al, al

*xor* cx, cx

*mov* ch, 13

*mov* cl, 0

*mov* dh, 13

*mov* dl, 14

*mov* bh, 0CEh

*int* 10h

*lea* dx, invalid

*mov* ah, 09h

*int* 21h

*call* DisplayContinue

*ret*

InvalidInputProc endp

DisplayContinue Proc

*lea* dx, continue

*mov* ah, 09h

*int* 21h

*mov* ah, 01h

*int* 21h

*ret*

DisplayContinue endp

ExitProgram:

*mov* ax, 4C00h

*int* 21h

end ProgramStart