VILLARAZO, JOHN KYLE A.

CS243 F4 LABORATORY HANDS-ON EXERCISES

**EXER12.ASM**

*; Filename: EXER12.ASM*

*; Programmer Name: JOHN KYLE VILLARAZO*

*; Date: 9/20/2024*

*; Description: This assembly language program will display multiple string variables on a single line.*

.model small

.stack 200h

.data

    String1 *db* 'Line1', '$' *; DB = Define Byte String1 variable and assign 'Line1' value;*

    String2 *db* 'Line2', '$'

    String3 *db* 'Line3', '$'

.code

ProgramStart:

*mov* ax, @data

*mov* ds, ax

*mov* ah, 9               *; DOS print string function*

*mov* dx, offset String1  *; 1st string to print*

*int* 21h                 *; invoke DOS to print string*

*mov* dx, offset String2  *; 2nd string to print*

*int* 21h                 *; invoke DOS to print string*

*mov* dx, offset String3  *; 3rd string to print*

*int* 21h

*mov* ah, 4ch             *; DOS terminate program function*

*int* 21h

END ProgramStart

**EXER13.ASM**

*; Filename: EXER13.ASM*

*; Programmer Name: JOHN KYLE VILLARAZO*

*; Date: 9/20/2024*

*; Description: This assembly language program will display multiple string variables on separate lines.*

.model small

.stack 200h

.data

    String1 *db* 'Line1', 0dh, 0ah, '$'

    String2 *db* 'Line2', 0dh, 0ah, '$'

    String3 *db* 'Line3', 0dh, 0ah, '$'

.code

ProgramStart:

*mov* ax, @data

*mov* ds, ax

*mov* ah, 9               *; DOS print string function*

*mov* dx, offset String1  *; 1st string to print*

*int* 21h                 *; invoke DOS to print string*

*mov* dx, offset String2

*int* 21h

*mov* dx, offset String3

*int* 21h

*mov* ah, 4ch

*int* 21h

end ProgramStart

**EXER14.ASM**

*; Filename: EXER14.ASM*

*; Programmer Name: JOHN KYLE VILLARAZO*

*; Date: 9/20/2024*

*; Description: This assembly language program will get character input and display back character input.*

.model small

.stack 200h

.data

    inputChar *db* ?      *; variable to store the input character*

    inputMsg *db* 'Enter a character: $'

    outputMsg *db* 'You entered: $'

.code

Main:

*; initialize the data segment*

*mov* ax, @data

*mov* ds, ax

*; display the input prompt message*

*mov* dx, offset inputMsg

*mov* ah, 09h

*int* 21h

*; read a character from the keyboard*

*mov* ah, 01h     *; function to read a character*

*int* 21h             *; call DOS interrupt*

*mov* inputChar, al   *; store the character in inputChar*

*; display the character back*

*mov* dl, inputChar   *; load the character to DL*

*mov* ah, 02h         *; function to display a character*

*int* 21h

*; exit program*

*mov* ax, 4C00h

*int* 21h

end Main

**EXER15.ASM**

*; Filename: EXER15.ASM*

*; Programmer Name: JOHN KYLE VILLARAZO*

*; Date: 9/20/2024*

*; Description: This assembly language program will continuously get character input and display back input until Enter key is pressed.*

.model small

.stack 200h

.data

.code

ProgramStart:

    EchoLoop:

*mov* ah, 1       *; DOS keyboard input function*

*int* 21h         *; get the next key*

*cmp* al, 13      *; was the key the Enter key?*

*jz* EchoDone     *; yes, so we're done echoing*

*mov* dl, al      *; put the character into DL*

*mov* ah, 2       *; DOS display output function*

*int* 21h

*jmp* EchoLoop    *; echo the next character*

    EchoDone:

*mov* ah, 4ch

*int* 21h

end ProgramStart

**EXER16.ASM**

*; Filename: EXER16.ASM*

*; Programmer Name: JOHN KYLE VILLARAZO*

*; Date: 9/20/2024*

*; Description: This assembly language program will get character input and determine if input is 'y' or 'Y' to display good morning message else display good afternoon message.*

.model small

.stack 200h

.data

    TimePrompt *db* 'Is it after 12 noon (Y/N)?$'

    GoodMorningMessage *label* byte

*db* 13, 10, 'Good morning, world!', 13, 10, '$'

    GoodAfternoonMessage *label* byte

*db* 13, 10, 'Good afternoon, world!', 13, 10, '$'

.code

ProgramStart:

*mov* ax, @data

*mov* ds, ax

*mov* dx, offset TimePrompt

*mov* ah, 9

*int* 21h

*mov* ah, 1           *; DOS get character function*

*int* 21h

*cmp* al, 'y'         *; typed lowercase 'y' for after noon?*

*jz* IsAfterNoon      *; jz = jump if zero. Yes, it's afternoon*

*cmp* al, 'Y'         *; typed uppercase 'Y' for afternoon?*

*jnz* IsMorning       *; no, it's before noon*

IsAfterNoon:

*mov* dx, offset GoodAfternoonMessage     *; point to the afternoon greeting*

*jmp* DisplayGreeting                     *; point to the afternoon greeting*

IsMorning:

*mov* dx, offset GoodMorningMessage       *; point to before noon greeting*

DisplayGreeting:

*mov* ah, 9           *; DOS print string function*

*int* 21h

*mov* ah, 4ch         *; DOS terminate program function*

*int* 21h

end ProgramStart

**EXER17.ASM**

*; Filename: EXER17.ASM*

*; Programmer Name: JOHN KYLE VILLARAZO*

*; Date: 9/20/2024*

*; Description: This assembly language program will get string in put and display back string.*

.model small

.stack 100h

.data

    MAXIMUM\_STRING\_LENGTH *EQU* 1000

    StringInput *db* MAXIMUM\_STRING\_LENGTH DUP (?)

    InputPrompt *db* 'Enter text: $'

.code

ProgramStart:

*mov* ax, @data

*mov* ds, ax

*mov* dx, offset InputPrompt

*mov* ah, 9

*int* 21h

*mov* ah, 3fh                     *; DOS read from handle function*

*mov* bx, 0                       *; standard input handle*

*mov* cx, MAXIMUM\_STRING\_LENGTH   *; read up to maximum number of characters*

*mov* dx, offset StringInput  *; Store the string here*

*int* 21h                     *; get the string*

*and* ax, ax                  *; were any character read?*

*jz* Done                     *; no, so you're done*

*mov* cx, ax                  *; put string length in CX you can use it as a counter*

*push* cx                     *; save the string length*

*mov* bx, offset StringInput

*pop* cx                      *; get back the string length*

*mov* ah, 40h                 *; DOS write form handle function*

*mov* bx, 1                   *; standard output handle*

*mov* dx, offset StringInput  *; prepare to print the string*

*int* 21h

Done:

*mov* ah, 4ch

*int* 21h

end ProgramStart

**EXER18.ASM**

*; Filename: EXER18.ASM*

*; Programmer Name: JOHN KYLE VILLARAZO*

*; Date: 9/20/2024*

*; Description: This assembly language program will get string input and display the reverse of the string.*

.model small

.stack 100h

.data

    MAXIMUM\_STRING\_LENGTH *EQU* 100

    StringToReverse *db* MAXIMUM\_STRING\_LENGTH dup (?)

    ReverseString *db* MAXIMUM\_STRING\_LENGTH dup (?)

.code

ProgramStart:

*mov* ax, @data

*mov* ds, ax          *; set DS to point to the data segment*

*mov* ah, 3fh         *; DOS read from handle function*

*mov* bx, 0           *; standard input handle*

*mov* cx, MAXIMUM\_STRING\_LENGTH   *; read up to maximum number of characters*

*mov* dx, offset StringToReverse      *; store the string here*

*int* 21h                 *; get the string*

*and* ax, ax              *; were any characters read?*

*jz* Done                 *; no, so you're done*

*mov* cx, ax              *; put string length in CX, where you can use it as a counter*

*push* cx                 *; save the string length*

*mov* bx, offset StringToReverse

*mov* si, offset ReverseString

*add* si, cx

*dec* si                  *; point to the end of the reverse string buffer*

ReverseLoop:

*mov* al, [bx]            *; get the next character*

*mov* [si], al            *; store the chracters in reverse order*

*inc* bx                  *; point to next character*

*dec* si                  *; point to previous location in reverse buffer*

*loop* ReverseLoop        *; move next character, if any*

*pop* cx                  *; get back the string length*

*mov* ah, 40h             *; DOS write from handle function*

*mov* bx, 1               *; standard output handle*

*mov* dx, offset ReverseString    *; print this string*

*int* 21h                         *; print the reversed string*

Done:

*mov* ah, 4ch             *; DOS terminate program function*

*int* 21h

end ProgramStart

**EXER19.ASM**

*; Filename: EXER19.ASM*

*; Programmer Name: JOHN KYLE VILLARAZO*

*; Date: 9/20/2024*

.model small

.stack 100h

.data

    firstChar *db* ?

    secondChar *db* ?

    thirdChar *db* ?

    firstMsg *db* 'Enter first character: $'

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

    thirdMsg *db*  0dh, 0ah, 'Enter third character: $'

    firstOutput *db*  0dh, 0ah, 'The first character is $'

    secondOutput *db*  0dh, 0ah, 'The second character is $'

    thirdOutput *db*  0dh, 0ah, 'The third character is $'

    dot *db* '.', 0dh, 0ah, '$'

.code

Main:

*; initialize the data segment*

*mov* ax, @data

*mov* ds, ax

*; display the input prompt message*

*mov* dx, offset firstMsg

*mov* ah, 09h

*int* 21h

*; read a character from the keyboard*

*mov* ah, 01h     *; function to read a character*

*int* 21h             *; call DOS interrupt*

*mov* firstChar, al   *; store the character in inputChar*

*mov* ah, 02h

*mov* dl, 0Ah

*int* 21h

*; display the input prompt message*

*mov* dx, offset secondMsg

*mov* ah, 09h

*int* 21h

*; read a character from the keyboard*

*mov* ah, 01h     *; function to read a character*

*int* 21h             *; call DOS interrupt*

*mov* secondChar, al   *; store the character in inputChar*

*mov* ah, 02h

*mov* dl, 0Ah

*int* 21h

*; display the input prompt message*

*mov* dx, offset thirdMsg

*mov* ah, 09h

*int* 21h

*; read a character from the keyboard*

*mov* ah, 01h     *; function to read a character*

*int* 21h             *; call DOS interrupt*

*mov* thirdChar, al   *; store the character in inputChar*

*mov* ah, 02h

*mov* dl, 0Ah

*int* 21h

*mov* dx, offset firstOutput

*mov* ah, 09h

*int* 21h

*; display the character back*

*mov* dl, firstChar   *; load the character to DL*

*mov* ah, 02h         *; function to display a character*

*int* 21h

*mov* dx, offset dot

*mov* ah, 09h

*int* 21h

*mov* dx, offset secondOutput

*mov* ah, 09h

*int* 21h

*mov* dl, secondChar   *; load the character to DL*

*mov* ah, 02h         *; function to display a character*

*int* 21h

*mov* dx, offset dot

*mov* ah, 09h

*int* 21h

*mov* dx, offset thirdOutput

*mov* ah, 09h

*int* 21h

*mov* dl, thirdChar   *; load the character to DL*

*mov* ah, 02h         *; function to display a character*

*int* 21h

*mov* dx, offset dot

*mov* ah, 09h

*int* 21h

*; exit program*

*mov* ax, 4C00h

*int* 21h

end Main

**EXER20.ASM**

*; Filename: EXER20.ASM*

*; Programmer Name: JOHN KYLE VILLARAZO*

*; Date: 9/20/2024*

.model small

.stack 100h

.data

    char *db* ?

    charMsg *db* 'Enter a character: $'

    YesMsg *label* byte

*db* 0dh, 0ah, 13, 10, 'Yes, you have entered letter A.', 13, 10, '$'

    NoMsg *label* byte

*db* 0dh, 0ah, 13, 10, 'No, you have not entered letter A. You entered character $'

    dot *db* '.$'

.code

Main:

*mov* ax, @data

*mov* ds, ax

*mov* dx, offset charMsg

*mov* ah, 9

*int* 21h

*; read a character from the keyboard*

*mov* ah, 01h     *; function to read a character*

*int* 21h             *; call DOS interrupt*

*mov* char, al   *; store the character in inputChar*

*cmp* al, 'a'

*jz* IsYes

*cmp* al, 'A'

*jnz* IsNo

IsYes:

*mov* dx, offset YesMsg

*jmp* DisplayAll

IsNo:

*mov* dx, offset NoMsg

*mov* ah, 09h

*int* 21h

*mov* dl, char   *; load the character to DL*

*mov* ah, 02h         *; function to display a character*

*int* 21h

*mov* dx, offset dot

DisplayAll:

*mov* ah, 9           *; DOS print string function*

*int* 21h

*mov* ah, 4ch         *; DOS terminate program function*

*int* 21h

end Main

**EXER21.ASM**

*; Filename: EXER21.ASM*

*; Programmer Name: JOHN KYLE VILLARAZO*

*; Date: 9/20/2024*

.model small

.stack 200h

.data

    firstName *db* 20, 20 dup(?)

    middleName *db* 20, 20 dup(?)

    familyName *db* 20, 20 dup(?)

    regMsg *db* 0dh, 0ah, 'REGISTRATION FORM$', 0dh, 0ah

    firstNamePrompt *db* 0Dh, 0Ah,'Enter your first name: $'

    middleNamePrompt *db* 0Dh, 0Ah, 'Enter your middle name: $'

    familyNamePrompt *db* 0Dh, 0Ah, 'Enter your family name: $'

    greetingMessage *db* 0Dh, 0Ah, 'Hello, $'

    space *db* ' $'

    exclamation *db* '!$'

.code

Main:

*mov* ax, @data

*mov* ds, ax

*mov* dx, OFFSET regMsg

*mov* ah, 09h

*int* 21h

*mov* dx, OFFSET firstNamePrompt

*mov* ah, 09h

*int* 21h

*lea* dx, firstName

*mov* ah, 0Ah

*int* 21h

*mov* dx, OFFSET middleNamePrompt

*mov* ah, 09h

*int* 21h

*lea* dx, middleName

*mov* ah, 0Ah

*int* 21h

*mov* dx, OFFSET familyNamePrompt

*mov* ah, 09h

*int* 21h

*lea* dx, familyName

*mov* ah, 0Ah

*int* 21h

*mov* dx, OFFSET greetingMessage

*mov* ah, 09h

*int* 21h

*lea* si, firstName + 2

*mov* cl, [firstName+1]

DisplayFirstName:

*mov* dl, [si]

*mov* ah, 02h

*int* 21h

*inc* si

*loop* DisplayFirstName

*mov* dl, ' '

*mov* ah, 02h

*int* 21h

*lea* si, middleName + 2

*mov* cl, [middleName+1]

DisplayMiddleName:

*mov* dl, [si]

*mov* ah, 02h

*int* 21h

*inc* si

*loop* DisplayMiddleName

*mov* dl, ' '

*mov* ah, 02h

*int* 21h

*lea* si, familyName + 2

*mov* cl, [familyName+1]

DisplayFamilyName:

*mov* dl, [si]

*mov* ah, 02h

*int* 21h

*inc* si

*loop* DisplayFamilyName

*mov* dl, '!'

*mov* ah, 02h

*int* 21h

*; End program*

*mov* ax, 4C00h

*int* 21h

END Main