JOHN ZILLION C. REYES

CS243 F1 LABORATORY

COMPILATION OF HANDS-ON EXERCISES AND HANDS-ON EXAMS

MIDTERM

First Semester SY 2024-2025

DATE: October 4, 2024

1. Hand-on Exercises
2. EXER1.ASM

; Filename: Exer1.asm

; Displaying single-characters, numbers, and symbols

; Programmer Name: JOHN ZILLION REYES

; Date: August 30, 2024

.model small

.data

    szTitle db "Filename: Exer1.asm", 0Ah, "Displaying single-characters, numbers, and symbols", 0Ah, "Programmer Name: JOHN ZILLION REYES", 0Ah, "Date: August 30, 2024", 0Ah, 0Ah, '$'

.code

.stack 100

start:

    mov ax, @data

    mov ds, ax

    lea dx, szTitle

    mov ah, 09h

    int 21h

    mov ah, 02h

    mov dl, 61h

    int 21h;

    mov ah, 02h

    mov dl, 62h

    int 21h;

    mov ah, 02h

    mov dl, 63h

    int 21h;

    mov ah, 02h

    mov dl, 64h

    int 21h;

    mov ah, 02h

    mov dl, 65h

    int 21h;

    mov ah, 02h

    mov dl, 66h

    int 21h;

    mov ah, 02h

    mov dl, 67h

    int 21h;

    mov ah, 02h

    mov dl, 68h

    int 21h;

    mov ah, 02h

    mov dl, 69h

    int 21h;

    mov ah, 02h

    mov dl, 6Ah

    int 21h;

    mov ah, 02h

    mov dl, 6Bh

    int 21h;

    mov ah, 02h

    mov dl, 6Ch

    int 21h;

    mov ah, 02h

    mov dl, 6Dh

    int 21h;

    mov ah, 02h

    mov dl, 6Eh

    int 21h;

    mov ah, 02h

    mov dl, 6Fh

    int 21h;

    mov ah, 02h

    mov dl, 70h

    int 21h;

    mov ah, 02h

    mov dl, 71h

    int 21h;

    mov ah, 02h

    mov dl, 72h

    int 21h;

    mov ah, 02h

    mov dl, 73h

    int 21h;

    mov ah, 02h

    mov dl, 74h

    int 21h;

    mov ah, 02h

    mov dl, 75h

    int 21h;

    mov ah, 02h

    mov dl, 76h

    int 21h;

    mov ah, 02h

    mov dl, 77h

    int 21h;

    mov ah, 02h

    mov dl, 78h

    int 21h;

    mov ah, 02h

    mov dl, 79h

    int 21h;

    mov ah, 02h

    mov dl, 7Ah

    int 21h;

    mov ah, 02h

    mov dl, 0Ah

    int 21h;

;

    mov ah, 02h

    mov dl, 41h

    int 21h;

    mov ah, 02h

    mov dl, 20h

    int 21h;

    mov ah, 02h

    mov dl, 42h

    int 21h;

    mov ah, 02h

    mov dl, 20h

    int 21h;

    mov ah, 02h

    mov dl, 43h

    int 21h;

    mov ah, 02h

    mov dl, 20h

    int 21h;

    mov ah, 02h

    mov dl, 44h

    int 21h;

    mov ah, 02h

    mov dl, 20h

    int 21h;

    mov ah, 02h

    mov dl, 45h

    int 21h;

    mov ah, 02h

    mov dl, 20h

    int 21h;

    mov ah, 02h

    mov dl, 46h

    int 21h;

    mov ah, 02h

    mov dl, 20h

    int 21h;

    mov ah, 02h

    mov dl, 47h

    int 21h;

    mov ah, 02h

    mov dl, 20h

    int 21h;

    mov ah, 02h

    mov dl, 48h

    int 21h;

    mov ah, 02h

    mov dl, 20h

    int 21h;

    mov ah, 02h

    mov dl, 49h

    int 21h;

    mov ah, 02h

    mov dl, 20h

    int 21h;

    mov ah, 02h

    mov dl, 4Ah

    int 21h;

    mov ah, 02h

    mov dl, 20h

    int 21h;

    mov ah, 02h

    mov dl, 4Bh

    int 21h;

    mov ah, 02h

    mov dl, 20h

    int 21h;

    mov ah, 02h

    mov dl, 4Ch

    int 21h;

    mov ah, 02h

    mov dl, 20h

    int 21h;

    mov ah, 02h

    mov dl, 4Dh

    int 21h;

    mov ah, 02h

    mov dl, 20h

    int 21h;

    mov ah, 02h

    mov dl, 4Eh

    int 21h;

    mov ah, 02h

    mov dl, 20h

    int 21h;

    mov ah, 02h

    mov dl, 4Fh

    int 21h;

    mov ah, 02h

    mov dl, 20h

    int 21h;

    mov ah, 02h

    mov dl, 50h

    int 21h;

    mov ah, 02h

    mov dl, 20h

    int 21h;

    mov ah, 02h

    mov dl, 51h

    int 21h;

    mov ah, 02h

    mov dl, 20h

    int 21h;

    mov ah, 02h

    mov dl, 52h

    int 21h;

    mov ah, 02h

    mov dl, 20h

    int 21h;

    mov ah, 02h

    mov dl, 53h

    int 21h;

    mov ah, 02h

    mov dl, 20h

    int 21h;

    mov ah, 02h

    mov dl, 54h

    int 21h;

    mov ah, 02h

    mov dl, 20h

    int 21h;

    mov ah, 02h

    mov dl, 55h

    int 21h;

    mov ah, 02h

    mov dl, 20h

    int 21h;

    mov ah, 02h

    mov dl, 56h

    int 21h;

    mov ah, 02h

    mov dl, 20h

    int 21h;

    mov ah, 02h

    mov dl, 57h

    int 21h;

    mov ah, 02h

    mov dl, 20h

    int 21h;

    mov ah, 02h

    mov dl, 58h

    int 21h;

    mov ah, 02h

    mov dl, 20h

    int 21h;

    mov ah, 02h

    mov dl, 59h

    int 21h;

    mov ah, 02h

    mov dl, 20h

    int 21h;

    mov ah, 02h

    mov dl, 5Ah

    int 21h;

;

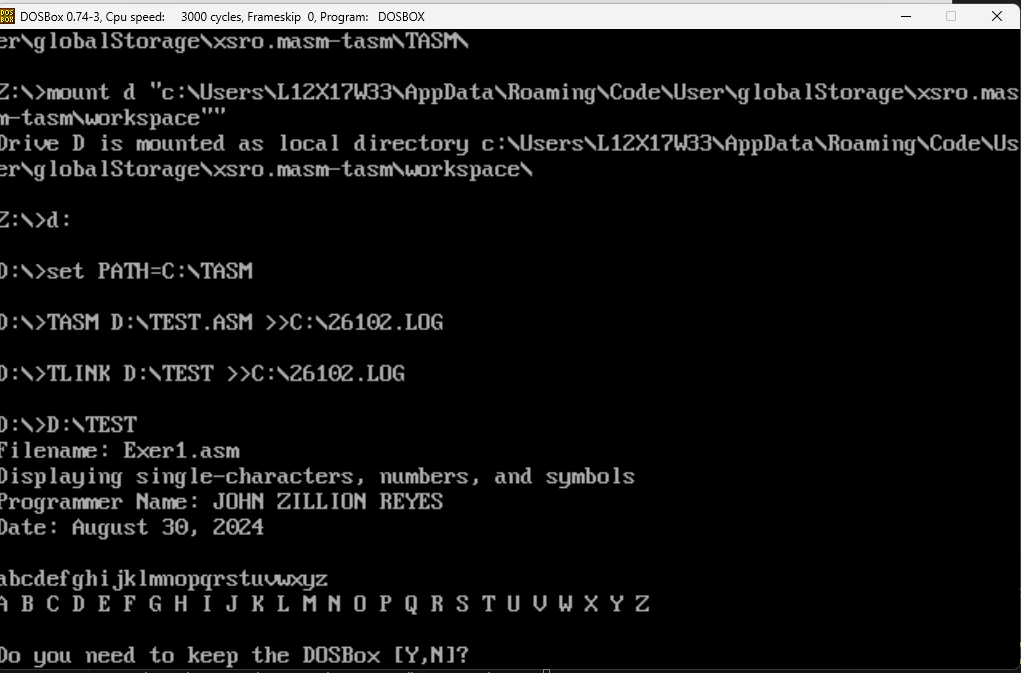
    mov ah, 02h

    mov dl, 0Ah

    int 21h;

int 27h

end start



1. EXER2.ASM

; Filename: Exer2.asm

; Displaying single-characters, numbers, and symbols

; Programmer Name: JOHN ZILLION REYES

; Date: August 30, 2024

.model small

.data

    szTitle db "Filename: Exer2.asm", 0Ah, "Displaying single-characters, numbers, and symbols", 0Ah, "Programmer Name: JOHN ZILLION REYES", 0Ah, "Date: August 30, 2024", 0Ah, 0Ah, '$'

.code

.stack 100

start:

    mov ax, @data

    mov ds, ax

    lea dx, szTitle

    mov ah, 09h

    int 21h

    mov ah, 02h

    mov dl, 74

    int 21h

    mov ah, 02h

    mov dl, 111

    int 21h

    mov ah, 02h

    mov dl, 104

    int 21h

    mov ah, 02h

    mov dl, 110

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 90

    int 21h

    mov ah, 02h

    mov dl, 105

    int 21h

    mov ah, 02h

    mov dl, 108

    int 21h

    mov ah, 02h

    mov dl, 108

    int 21h

    mov ah, 02h

    mov dl, 105

    int 21h

    mov ah, 02h

    mov dl, 111

    int 21h

    mov ah, 02h

    mov dl, 110

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 67

    int 21h

    mov ah, 02h

    mov dl, 97

    int 21h

    mov ah, 02h

    mov dl, 98

    int 21h

    mov ah, 02h

    mov dl, 97

    int 21h

    mov ah, 02h

    mov dl, 108

    int 21h

    mov ah, 02h

    mov dl, 108

    int 21h

    mov ah, 02h

    mov dl, 101

    int 21h

    mov ah, 02h

    mov dl, 114

    int 21h

    mov ah, 02h

    mov dl, 111

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 82

    int 21h

    mov ah, 02h

    mov dl, 101

    int 21h

    mov ah, 02h

    mov dl, 121

    int 21h

    mov ah, 02h

    mov dl, 101

    int 21h

    mov ah, 02h

    mov dl, 115

    int 21h

    mov ah, 02h

    mov dl, 10

    int 21h

    mov ah, 02h

    mov dl, 66

    int 21h

    mov ah, 02h

    mov dl, 97

    int 21h

    mov ah, 02h

    mov dl, 99

    int 21h

    mov ah, 02h

    mov dl, 104

    int 21h

    mov ah, 02h

    mov dl, 101

    int 21h

    mov ah, 02h

    mov dl, 108

    int 21h

    mov ah, 02h

    mov dl, 111

    int 21h

    mov ah, 02h

    mov dl, 114

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 111

    int 21h

    mov ah, 02h

    mov dl, 102

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 83

    int 21h

    mov ah, 02h

    mov dl, 99

    int 21h

    mov ah, 02h

    mov dl, 105

    int 21h

    mov ah, 02h

    mov dl, 101

    int 21h

    mov ah, 02h

    mov dl, 110

    int 21h

    mov ah, 02h

    mov dl, 99

    int 21h

    mov ah, 02h

    mov dl, 101

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 105

    int 21h

    mov ah, 02h

    mov dl, 110

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 67

    int 21h

    mov ah, 02h

    mov dl, 111

    int 21h

    mov ah, 02h

    mov dl, 109

    int 21h

    mov ah, 02h

    mov dl, 112

    int 21h

    mov ah, 02h

    mov dl, 117

    int 21h

    mov ah, 02h

    mov dl, 116

    int 21h

    mov ah, 02h

    mov dl, 101

    int 21h

    mov ah, 02h

    mov dl, 114

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 83

    int 21h

    mov ah, 02h

    mov dl, 99

    int 21h

    mov ah, 02h

    mov dl, 105

    int 21h

    mov ah, 02h

    mov dl, 101

    int 21h

    mov ah, 02h

    mov dl, 110

    int 21h

    mov ah, 02h

    mov dl, 99

    int 21h

    mov ah, 02h

    mov dl, 101

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 50

    int 21h

    mov ah, 02h

    mov dl, 110

    int 21h

    mov ah, 02h

    mov dl, 100

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 89

    int 21h

    mov ah, 02h

    mov dl, 101

    int 21h

    mov ah, 02h

    mov dl, 97

    int 21h

    mov ah, 02h

    mov dl, 114

    int 21h

    mov ah, 02h

    mov dl, 10

    int 21h

    mov ah, 02h

    mov dl, 67

    int 21h

    mov ah, 02h

    mov dl, 111

    int 21h

    mov ah, 02h

    mov dl, 108

    int 21h

    mov ah, 02h

    mov dl, 108

    int 21h

    mov ah, 02h

    mov dl, 101

    int 21h

    mov ah, 02h

    mov dl, 103

    int 21h

    mov ah, 02h

    mov dl, 101

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 111

    int 21h

    mov ah, 02h

    mov dl, 102

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 67

    int 21h

    mov ah, 02h

    mov dl, 111

    int 21h

    mov ah, 02h

    mov dl, 109

    int 21h

    mov ah, 02h

    mov dl, 112

    int 21h

    mov ah, 02h

    mov dl, 117

    int 21h

    mov ah, 02h

    mov dl, 116

    int 21h

    mov ah, 02h

    mov dl, 101

    int 21h

    mov ah, 02h

    mov dl, 114

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 83

    int 21h

    mov ah, 02h

    mov dl, 116

    int 21h

    mov ah, 02h

    mov dl, 117

    int 21h

    mov ah, 02h

    mov dl, 100

    int 21h

    mov ah, 02h

    mov dl, 105

    int 21h

    mov ah, 02h

    mov dl, 101

    int 21h

    mov ah, 02h

    mov dl, 115

    int 21h

    mov ah, 02h

    mov dl, 10

    int 21h

    mov ah, 02h

    mov dl, 67

    int 21h

    mov ah, 02h

    mov dl, 101

    int 21h

    mov ah, 02h

    mov dl, 98

    int 21h

    mov ah, 02h

    mov dl, 117

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 73

    int 21h

    mov ah, 02h

    mov dl, 110

    int 21h

    mov ah, 02h

    mov dl, 115

    int 21h

    mov ah, 02h

    mov dl, 116

    int 21h

    mov ah, 02h

    mov dl, 105

    int 21h

    mov ah, 02h

    mov dl, 116

    int 21h

    mov ah, 02h

    mov dl, 117

    int 21h

    mov ah, 02h

    mov dl, 116

    int 21h

    mov ah, 02h

    mov dl, 101

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 111

    int 21h

    mov ah, 02h

    mov dl, 102

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 84

    int 21h

    mov ah, 02h

    mov dl, 101

    int 21h

    mov ah, 02h

    mov dl, 99

    int 21h

    mov ah, 02h

    mov dl, 104

    int 21h

    mov ah, 02h

    mov dl, 110

    int 21h

    mov ah, 02h

    mov dl, 111

    int 21h

    mov ah, 02h

    mov dl, 108

    int 21h

    mov ah, 02h

    mov dl, 111

    int 21h

    mov ah, 02h

    mov dl, 103

    int 21h

    mov ah, 02h

    mov dl, 121

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 45

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 85

    int 21h

    mov ah, 02h

    mov dl, 110

    int 21h

    mov ah, 02h

    mov dl, 105

    int 21h

    mov ah, 02h

    mov dl, 118

    int 21h

    mov ah, 02h

    mov dl, 101

    int 21h

    mov ah, 02h

    mov dl, 114

    int 21h

    mov ah, 02h

    mov dl, 115

    int 21h

    mov ah, 02h

    mov dl, 105

    int 21h

    mov ah, 02h

    mov dl, 116

    int 21h

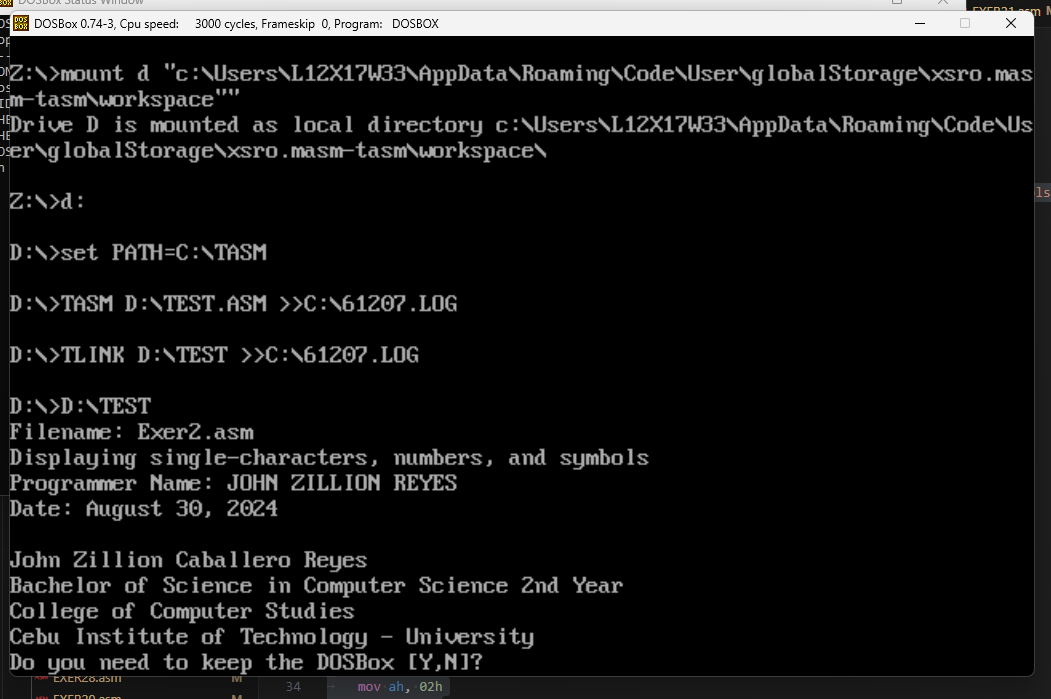
    mov ah, 02h

    mov dl, 121

    int 21h

int 27h

end start



1. EXER3.ASM

; Filename: Exer3.asm

; Displaying single-characters, numbers, and symbols

; Programmer Name: JOHN ZILLION REYES

; Date: August 30, 2024

.model small

.data

    szTitle db "Filename: Exer3.asm", 0Ah, "Displaying single-characters, numbers, and symbols", 0Ah, "Programmer Name: JOHN ZILLION REYES", 0Ah, "Date: August 30, 2024", 0Ah, 0Ah, '$'

.code

.stack 100

start:

    mov ax, @data

    mov ds, ax

    lea dx, szTitle

    mov ah, 09h

    int 21h

    mov ah, 02h

    mov dl, 48

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 49

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 50

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 51

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 52

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 53

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 54

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 55

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 56

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 57

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 10

    int 21h

    mov ah, 02h

    mov dl, 33

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 42

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 35

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 36

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 37

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 94

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 38

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 42

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 40

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 41

    int 21h

    mov ah, 02h

    mov dl, 10

    int 21h

    mov ah, 02h

    mov dl, 45

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 95

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 43

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 61

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 123

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 125

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 91

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 93

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 60

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

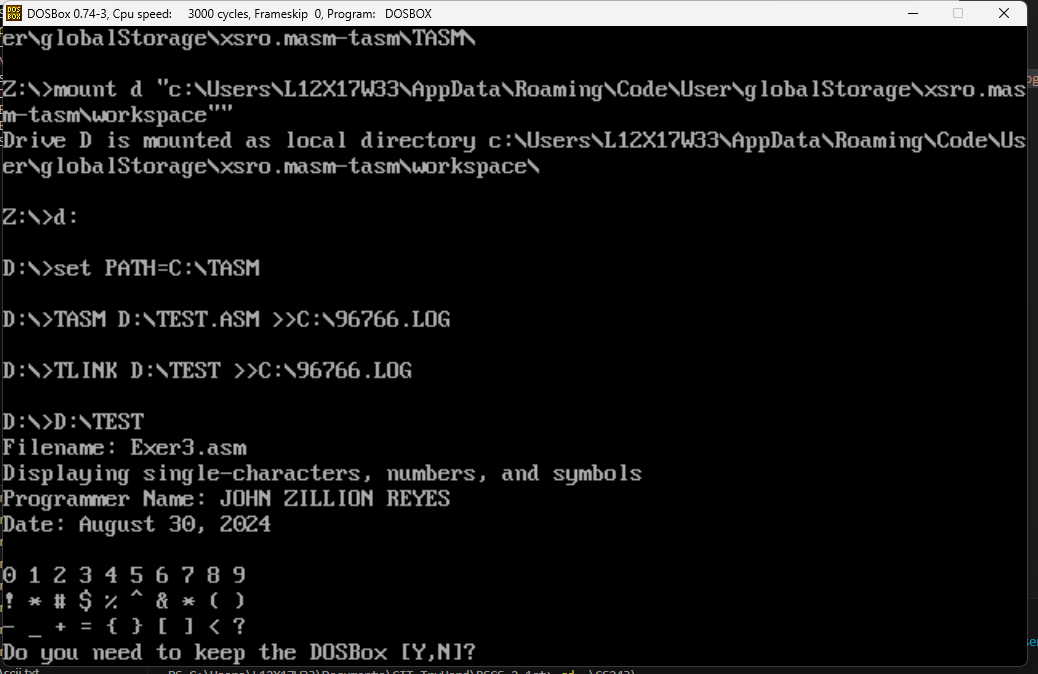
    mov ah, 02h

    mov dl, 63

    int 21h

int 27h

end start



1. EXER4.ASM

; Filename: Exer4.asm

; Displaying single-characters, numbers, and symbols

; Programmer Name: JOHN ZILLION REYES

; Date: August 30, 2024

.model small

.data

    szTitle db "Filename: Exer4.asm", 0Ah, "Displaying single-characters, numbers, and symbols", 0Ah, "Programmer Name: JOHN ZILLION REYES", 0Ah, "Date: August 30, 2024", 0Ah, 0Ah, '$'

.code

.stack 100

start:

    mov ax, @data

    mov ds, ax

    lea dx, szTitle

    mov ah, 09h

    int 21h

    mov ah, 09h

    mov bl, 84h

    mov cx, 7

    int 10h;

    mov ah, 02h

    mov dl, 67

    int 21h

    mov ah, 02h

    mov dl, 111

    int 21h

    mov ah, 02h

    mov dl, 108

    int 21h

    mov ah, 02h

    mov dl, 108

    int 21h

    mov ah, 02h

    mov dl, 101

    int 21h

    mov ah, 02h

    mov dl, 103

    int 21h

    mov ah, 02h

    mov dl, 101

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 10

    int 21h

    mov ah, 09h

    mov bl, 1Ah

    mov cx, 2

    int 10h;

    mov ah, 02h

    mov dl, 111

    int 21h

    mov ah, 02h

    mov dl, 102

    int 21h

    mov ah, 02h

    mov dl, 10

    int 21h

    mov ah, 09h

    mov bl, 3Ah

    mov cx, 8

    int 10h;

    mov ah, 02h

    mov dl, 67

    int 21h

    mov ah, 02h

    mov dl, 111

    int 21h

    mov ah, 02h

    mov dl, 109

    int 21h

    mov ah, 02h

    mov dl, 112

    int 21h

    mov ah, 02h

    mov dl, 117

    int 21h

    mov ah, 02h

    mov dl, 116

    int 21h

    mov ah, 02h

    mov dl, 101

    int 21h

    mov ah, 02h

    mov dl, 114

    int 21h

    mov ah, 02h

    mov dl, 10

    int 21h

    mov ah, 09h

    mov bl, 1Eh

    mov cx, 7

    int 10h;

    mov ah, 02h

    mov dl, 83

    int 21h

    mov ah, 02h

    mov dl, 116

    int 21h

    mov ah, 02h

    mov dl, 117

    int 21h

    mov ah, 02h

    mov dl, 100

    int 21h

    mov ah, 02h

    mov dl, 105

    int 21h

    mov ah, 02h

    mov dl, 101

    int 21h

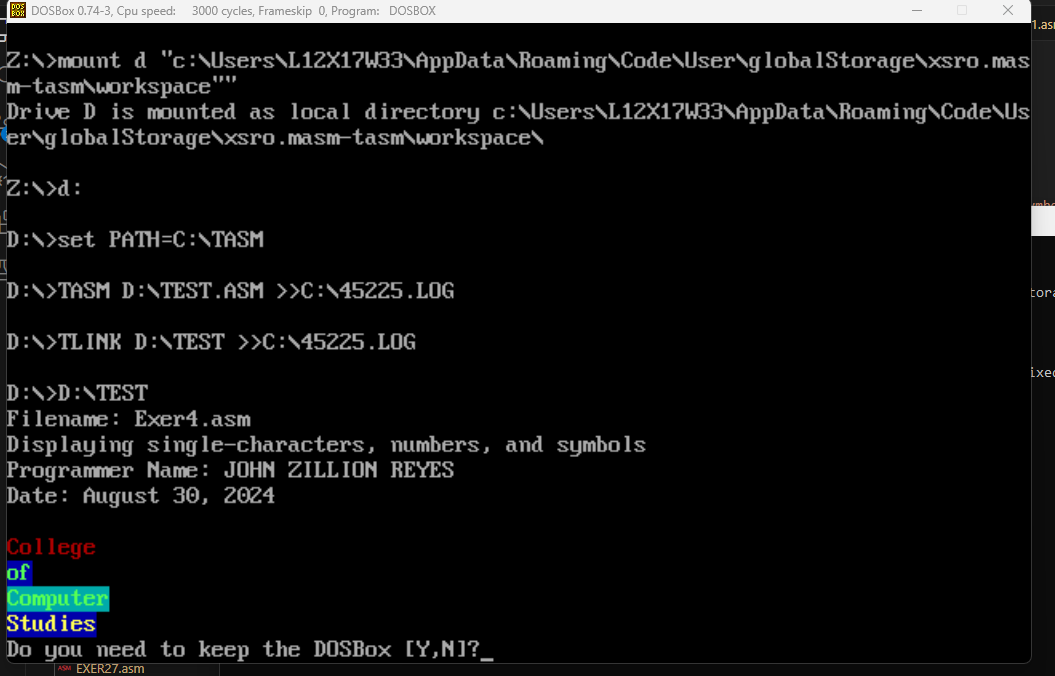
    mov ah, 02h

    mov dl, 115

    int 21h

int 27h

end start



1. EXER5.ASM

; Filename: Exer5.asm

; Displaying single-characters, numbers, and symbols

; Programmer Name: JOHN ZILLION REYES

; Date: August 30, 2024

.model small

.data

    szTitle db "Filename: Exer5.asm", 0Ah, "Displaying single-characters, numbers, and symbols", 0Ah, "Programmer Name: JOHN ZILLION REYES", 0Ah, "Date: August 30, 2024", 0Ah, 0Ah, '$'

.code

.stack 100

start:

    mov ax, @data

    mov ds, ax

    lea dx, szTitle

    mov ah, 09h

    int 21h

    mov ah, 09h

    mov bl, 10h

    mov cx, 5

    int 10h;

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 10

    int 21h

    mov ah, 09h

    mov bl, 10h

    mov cx, 2

    int 10h;

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 09h

    mov bl, 40h

    mov cx, 1

    int 10h;

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 09h

    mov bl, 10h

    mov cx, 2

    int 10h;

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 10

    int 21h

    mov ah, 09h

    mov bl, 10h

    mov cx, 1

    int 10h;

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 09h

    mov bl, 40h

    mov cx, 3

    int 10h;

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 09h

    mov bl, 10h

    mov cx, 1

    int 10h;

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 10

    int 21h

    mov ah, 09h

    mov bl, 10h

    mov cx, 2

    int 10h;

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 09h

    mov bl, 40h

    mov cx, 1

    int 10h;

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 09h

    mov bl, 10h

    mov cx, 2

    int 10h;

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 10

    int 21h

    mov ah, 09h

    mov bl, 10h

    mov cx, 5

    int 10h;

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

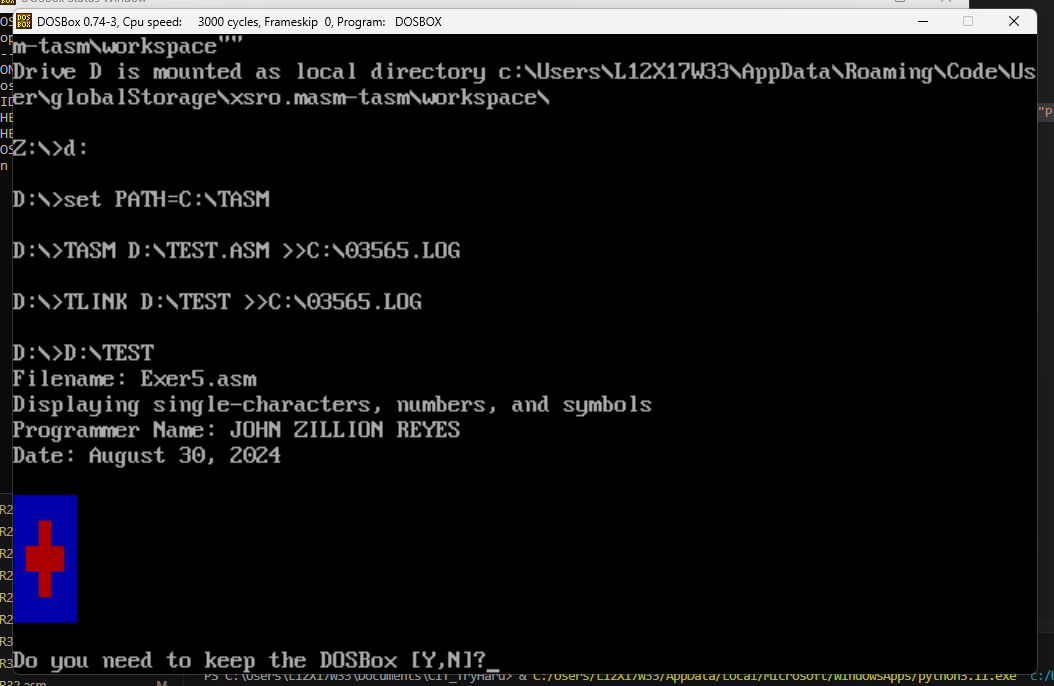
    mov ah, 02h

    mov dl, 10

    int 21h

int 27h

end start



1. EXER6.ASM

; Filename: Exer6.asm

; Displaying single-characters, numbers, and symbols

; Programmer Name: JOHN ZILLION REYES

; Date: August 30, 2024

.model small

.data

    szTitle db "Filename: Exer6.asm", 0Ah, "Displaying single-characters, numbers, and symbols", 0Ah, "Programmer Name: JOHN ZILLION REYES", 0Ah, "Date: August 30, 2024", 0Ah, 0Ah, '$'

.code

.stack 100

start:

    mov ax, @data

    mov ds, ax

    lea dx, szTitle

    mov ah, 09h

    int 21h

    mov ah, 09h

    mov bl, 06h

    mov cx, 1

    int 10h;

    mov ah, 02h

    mov dl, 73

    int 21h

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 09h

    mov bl, 84h

    mov cx, 1

    int 10h;

    mov ah, 02h

    mov dl, 03h

    int 21h;

    mov ah, 02h

    mov dl, 32

    int 21h

    mov ah, 09h

    mov bl, 06h

    mov cx, 1

    int 10h;

    mov ah, 02h

    mov dl, 85

    int 21h

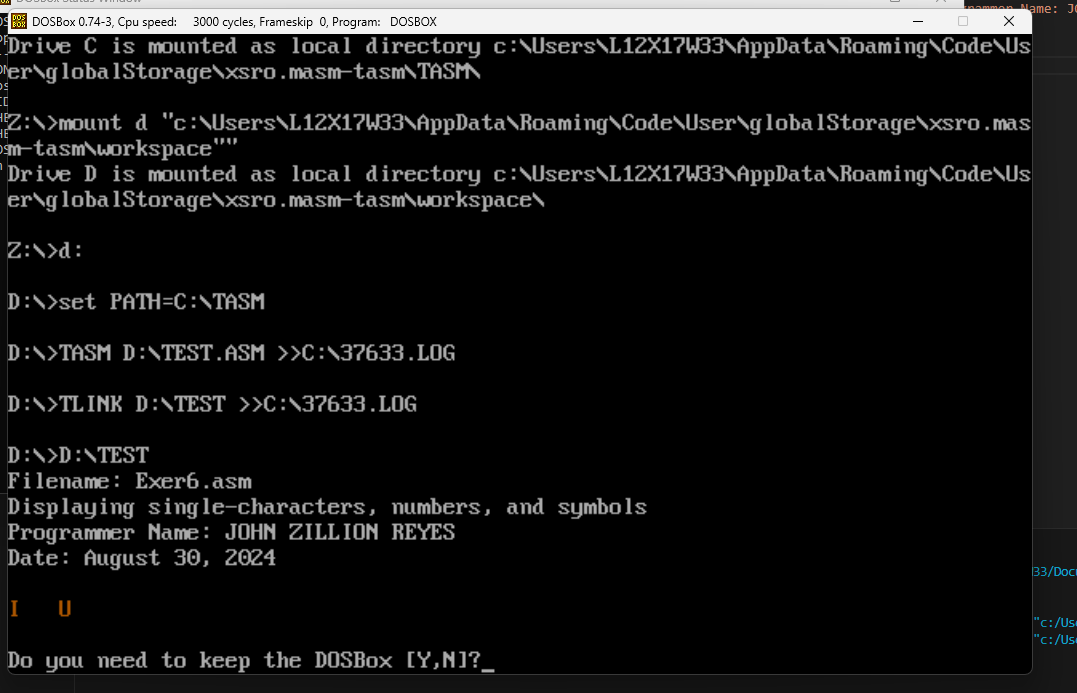
    mov ah, 02h

    mov dl, 10

    int 21h

int 27h

end start



1. EXER7.ASM

; Filename: Exer7.asm

; create your own multicolored character with blinking parts

; Programmer Name: JOHN ZILLION REYES

; Date: August 30, 2024

.model small

.stack 100h

.data

    szTitle db "Filename: Exer7.asm", 0Ah, "create your own multicolored character with blinking parts", 0Ah, "Programmer Name: JOHN ZILLION REYES", 0Ah, "Date: August 30, 2024", 0Ah, 0Ah, '$'

.code

start:

    mov ax, @data

    mov ds, ax

    lea dx, szTitle

    mov ah, 09h

    int 21h

    ; line 1

    mov ah, 02h

    mov dl, 0ah  ;newline

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 09h

    mov bl, 11h ; color blue

    mov cx, 4

    int 10h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    ; line 2

    mov ah, 02h

    mov dl, 0ah  ;newline

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 09h

    mov bl, 11h ; color blue

    mov cx, 6

    int 10h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    ; line 3

    mov ah, 02h

    mov dl, 0ah  ;newline

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 09h

    mov bl, 22h ; beak green

    mov cx, 1

    int 10h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 09h

    mov bl, 11h ; color blue

    mov cx, 1

    int 10h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 09h

    mov bl, 87h ; blink

    mov cx, 1

    int 10h

    mov ah, 02h

    mov dl, 30h ; eye 9

    int 21h

    mov ah, 09h

    mov bl, 11h ; color blue

    mov cx, 3

    int 10h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    ; line 4

    mov ah, 02h

    mov dl, 0ah  ;newline

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 09h

    mov bl, 22h ; beak green

    mov cx, 3

    int 10h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 09h

    mov bl, 11h ; color blue

    mov cx, 5

    int 10h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    ; line 5

    mov ah, 02h

    mov dl, 0ah  ;newline

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 09h

    mov bl, 22h ; beak green

    mov cx, 3

    int 10h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 09h

    mov bl, 11h ; color blue

    mov cx, 3

    int 10h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    ; line 6

    mov ah, 02h

    mov dl, 0ah  ;newline

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 09h

    mov bl, 11h ; color blue

    mov cx, 3

    int 10h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 09h

    mov bl, 11h ; color blue

    mov cx, 3

    int 10h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 09h

    mov bl, 11h ; color blue

    mov cx, 1

    int 10h

    mov ah, 02h

    mov dl, 20h

    int 21h

    ; line 7

    mov ah, 02h

    mov dl, 0ah  ;newline

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 09h

    mov bl, 11h ; color blue

    mov cx, 8

    int 10h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 09h

    mov bl, 55h ; line purple

    mov cx, 1

    int 10h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 09h

    mov bl, 11h ; color blue

    mov cx, 1

    int 10h

    mov ah, 02h

    mov dl, 20h

    int 21h

    ; line 8

    mov ah, 02h

    mov dl, 0ah  ;newline

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 09h

    mov bl, 11h ; color blue

    mov cx, 4

    int 10h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 09h

    mov bl, 55h ; line purple

    mov cx, 1

    int 10h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 09h

    mov bl, 11h ; color blue

    mov cx, 3

    int 10h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 09h

    mov bl, 55h ; line purple

    mov cx, 1

    int 10h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 09h

    mov bl, 11h ; color blue

    mov cx, 2

    int 10h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    ; line 9

    mov ah, 02h

    mov dl, 0ah  ;newline

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 09h

    mov bl, 11h ; color blue

    mov cx, 5

    int 10h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 09h

    mov bl, 55h ; line purple

    mov cx, 3

    int 10h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 09h

    mov bl, 11h ; color blue

    mov cx, 3

    int 10h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    ; line 10

    mov ah, 02h

    mov dl, 0ah  ;newline

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 09h

    mov bl, 11h ; color blue

    mov cx, 11

    int 10h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    ; line 11

    mov ah, 02h

    mov dl, 0ah  ;newline

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 09h

    mov bl, 11h ; color blue

    mov cx, 9

    int 10h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    ; line 12

    mov ah, 02h

    mov dl, 0ah  ;newline

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 09h

    mov bl, 11h ; color blue

    mov cx, 7

    int 10h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    mov ah, 02h

    mov dl, 20h

    int 21h

    ; line 13

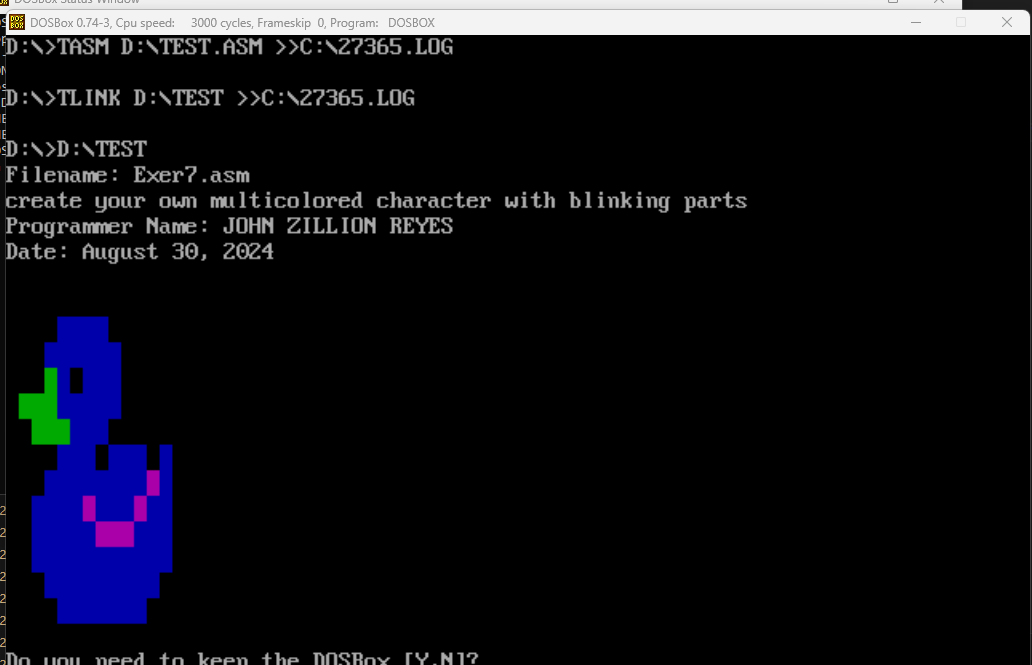
    mov ah, 02h

    mov dl, 0ah  ;newline

    int 21h

int 27h

end start



1. EXER8.ASM

; Filename: Exer8.asm

; Program Description: Display text using string variables (4 string variables)

; Programmer Name: JOHN ZILLION REYES

; Date: August 30, 2024

.model small

.stack 100h

.data

    szTitle db "Filename: Exer8.asm", 0Ah, "Program Description: Display text using string variables (4 string variables)", 0Ah, "Programmer Name: JOHN ZILLION REYES", 0Ah, "Date: August 30, 2024", 0Ah, 0Ah, '$'

    msg1 db "John Zillion C. Reyes$"

    msg2 db "Bachelor of Science in Computer Science 2nd Year$"

    msg3 db "College of Computer Studies$"

    msg4 db "Cebu Institute of Technology - University$"

.code

.startup

start:

    ; data

    mov ax, @data

    mov ds, ax

    lea dx, szTitle

    mov ah, 09h

    int 21h

    mov ah, 09h

    mov dx, offset msg1

    int 21h

    mov ah, 02h

    mov dl, 0ah  ;newline

    int 21h

    mov ah, 09h

    mov dx, offset msg2

    int 21h

    mov ah, 02h

    mov dl, 0ah  ;newline

    int 21h

    mov ah, 09h

    mov dx, offset msg3

    int 21h

    mov ah, 02h

    mov dl, 0ah  ;newline

    int 21h

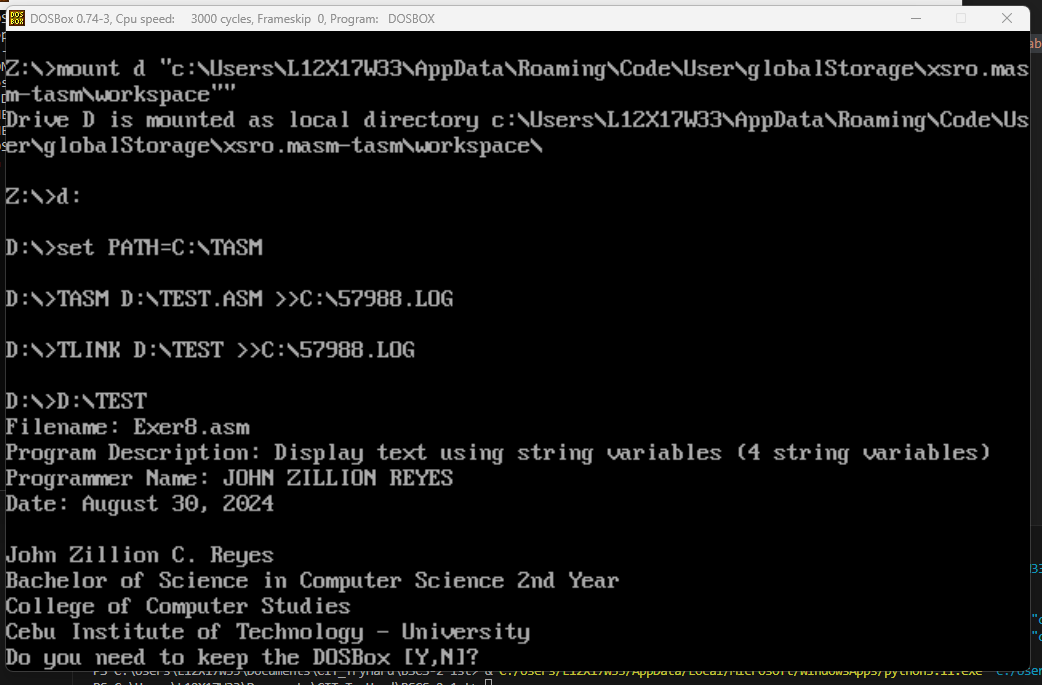
    mov ah, 09h

    mov dx, offset msg4

    int 21h

int 27h

end start



1. EXER9.ASM

; Filename: EXER9.ASM

; Programmer Name: JOHN ZILLION REYES

; Program Description: Display August Calendar

; Date: August 29, 2024

.model small

.stack 100h

.data

    szTitle db "Filename: EXER9.ASM", 0Ah, "Programmer Name: JOHN ZILLION REYES", 0Ah, "Program Description: Display August Calendar", 0Ah, "Date: August 29, 2024", 0Ah, 0Ah, '$'

    str1 db '|-----------------------------------------|', 13, 10

         db '|                 AUGUST                  |', 13, 10

         db '|-----------------------------------------|', 13, 10

         db '| MON | TUE | WED | THU | FRI | SAT | SUN |', 13, 10

         db '|-----------------------------------------|', 13, 10

         db '|   29|   30|   31|    1|    2|    3|    4|', 13, 10

         db '|-----------------------------------------|', 13, 10

         db '|    5|    6|    7|    8|    9|   10|   11|', 13, 10

         db '|-----------------------------------------|', 13, 10

         db '|   12|   13|   14|   15|   16|   17|   18|', 13, 10

         db '|-----------------------------------------|', 13, 10

         db '|   19|   20|   21|   22|   23|   24|   25|', 13, 10

         db '|-----------------------------------------|', 13, 10

         db '|   26|   27|   28|   29|   30|   31|    1|', 13, 10

         db '|-----------------------------------------|', 13, 10, '$'

.code

start:

    mov ax, @data

    mov ds, ax

    lea dx, szTitle

    mov ah, 09h

    int 21h

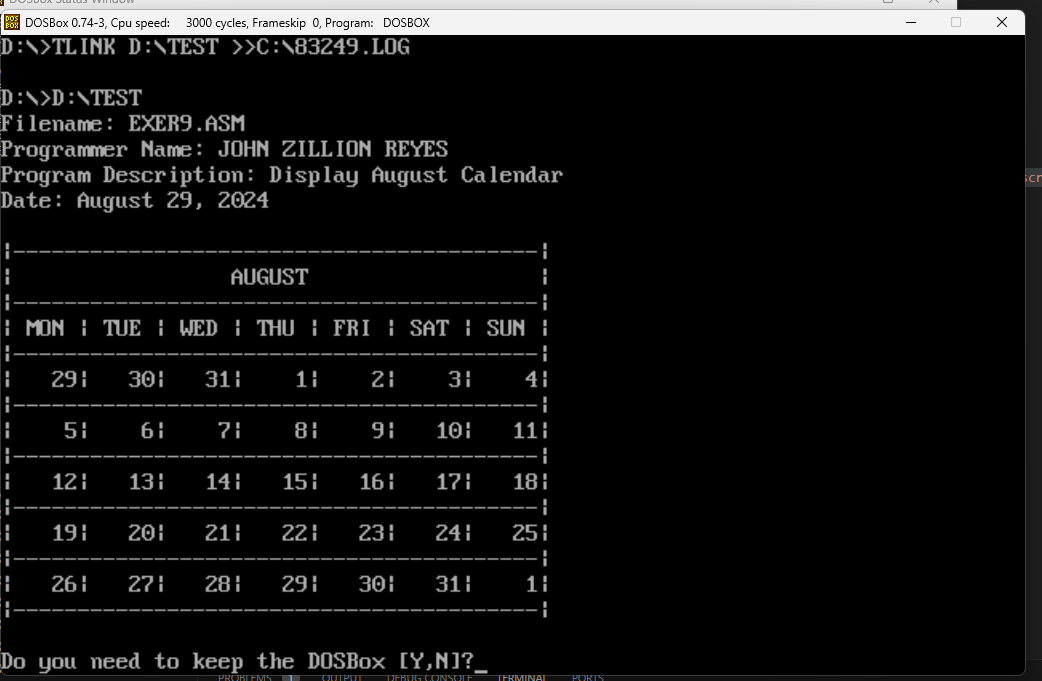
    lea dx, str1

    mov ah, 09h

    int 21h

int 27h

end start



1. EXER10.ASM

; Filename: Exer10.asm

; Program Description: Displaying single chraracters, numbers, & symbols

; Programmer Name: JOHN ZILLION REYES

; Date: August 29, 2024

.model small

.stack 100h

.data

    szTitle db "Filename: Exer10.asm", 0Ah, "Program Description: Displaying single chraracters, numbers, & symbols", 0Ah, "Programmer Name: JOHN ZILLION REYES", 0Ah, "Date: August 29, 2024", 0Ah, 0Ah, '$'

    ;box shapes

    boxtopL db 218, "$"

    boxtopR db 191, "$"

    boxbotL db 192, "$"

    boxbotR db 217, "$"

    boxcen db 196, "$"

    ; title

    form db "College Enrollment Form$"

    college db "Cebu Institute of Technology - University $"

    ; Field Names (length set to 20)

    FullName db  "     Full Name      $"

    Address db   "     Address        $"

    BirthDate db "     Birth Date     $"

    Gender db    "     Gender         $"

    StuNum db    "     Student Number $"

    StuMail db   "     Student Email  $"

    Comp db      "     Company        $"

    Course db    "     Courses        $"

    Comment db   "     Addt. Comments $"

    Date db      "Date: $"

    Sig db "Signature$"

    ; Field Labels

.code

;description

main PROC

    ; code setup

    mov ax, @data

    mov ds, ax

    lea dx, szTitle

    mov ah, 09h

    int 21h

    mov ah, 00

    mov al, 03

    int 10h

    mov ah, 09

    mov bh, 00

    mov al, 20h

    mov cx, 800h ; bg

    mov bl, 70h

    int 10h

    ; code start

    call newline

    call banner

    ; FULL NAME

    mov ah, 09h

    mov dx, offset FullName

    int 21h

    mov cx, 20

    call boxtop

    call space

    call space

    mov cx, 20

    call boxtop

    call newline

    mov cx, 20

    call spaceloop

    mov cx, 20

    call boxbot

    call space

    call space

    mov cx, 20

    call boxbot

    ; ADDRESS

    call newline

    mov ah, 09h

    mov dx, offset Address

    int 21h

    mov cx, 44

    call boxtop ; box addre 1 big

    call newline

    mov cx, 20

    call spaceloop

    mov cx, 44

    call boxbot

    call newline ; box addre 2 small

    mov cx, 20

    call spaceloop

    mov cx, 20

    call boxtop

    call space

    call space

    mov cx, 20

    call boxtop

    call newline

    mov cx, 20

    call spaceloop

    mov cx, 20

    call boxbot

    call space

    call space

    mov cx, 20

    call boxbot

    ; Birth Date

    call newline

    mov ah, 09h

    mov dx, offset BirthDate

    int 21h

    mov cx, 12 ; topside 3 bd bop

    call boxtop

    call space

    call space

    mov cx, 12

    call boxtop

    call space

    call space

    mov cx, 12

    call boxtop

    call newline

    mov cx, 20

    call spaceloop

    mov cx, 12 ; botside 3 bd bop

    call boxbot

    call space

    call space

    mov cx, 12

    call boxbot

    call space

    call space

    mov cx, 12

    call boxbot

    call newline

    mov ax, 4c00h ; return 0

    int 21h

main ENDP

;top box side

boxtop PROC ; set cx before calling (min=1)

    ;Topside

    mov ah, 09h

    mov dx, offset boxtopL

    int 21h

    boxlength1:

    mov ah, 09h

    mov dx, offset boxcen

    int 21h

    loop boxlength1

    mov ah, 09h

    mov dx, offset boxtopR

    int 21h

    ret

boxtop ENDP

;bot box side

boxbot PROC ; set cx before calling (min=1)

    ;Botside

    mov ah, 09h

    mov dx, offset boxbotL

    int 21h

    boxlength2:

    mov ah, 09h

    mov dx, offset boxcen

    int 21h

    loop boxlength2

    mov ah, 09h

    mov dx, offset boxbotR

    int 21h

    ret

boxbot ENDP

; newline gen

newline PROC

    mov ah, 02h

    mov dl, 0ah  ;newline

    int 21h

    ret

newline ENDP

; space gen (1 space)

space PROC

    mov ah, 02h

    mov dl, 20h     ; space

    int 21h

    ret

space ENDP

;many space gen

spaceloop PROC

    spacelooper:

    call space

    loop spacelooper

    ret

spaceloop ENDP

;description

banner PROC

    mov ah, 09h

    mov bl, 43h     ; red

    mov cx, 80

    int 10h

    call newline

    mov ah, 09h

    mov bl, 43h     ; red

    mov cx, 80

    int 10h

    mov cx, 5

    call spaceloop

    mov ah, 09h

    mov dx, offset form

    int 21h

    mov cx, 5

    call spaceloop

    mov ah, 09h

    mov dx, offset college

    int 21h

    mov cx, 5

    call spaceloop

    call newline

    mov ah, 09h

    mov bl, 43h     ; red

    mov cx, 80

    int 10h

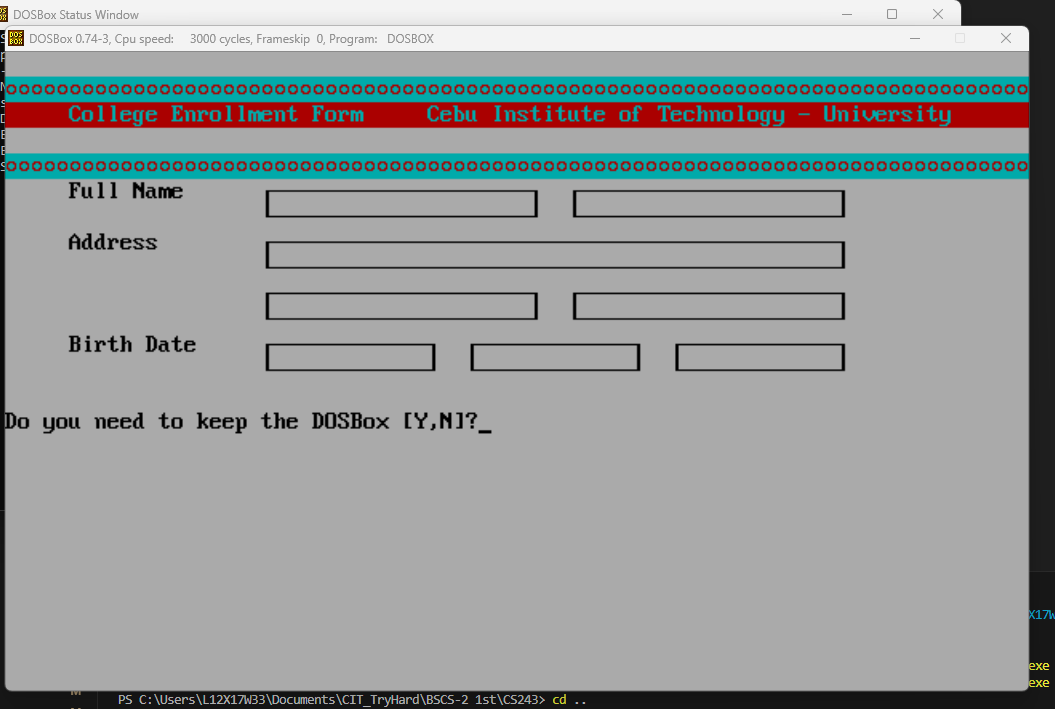
    call newline

    ret

    int 27h

banner ENDP

end main



1. EXER11.ASM

; Filename: EXER11.asm

; Program Description: Simulates a bank

; Programmer Name: JOHN ZILLION REYES

; Date Finished: September 13, 2024

.MODEL SMALL

.STACK 100

.DATA

    szTitle db "Filename: EXER11.asm", 0Ah, "Program Description: Simulates a bank", 0Ah, "Programmer Name: JOHN ZILLION REYES", 0Ah, "Date Finished: September 13, 2024", 0Ah, 0Ah, '$'

TEXT    db '   Log Out '

        db 13, 10, 10, '                        Dear Customers, Welcome to Bank X             ', 13, 10, 10

        db 13, 10, 10, '        <    Withdrawal                                   Payment    >', 13, 10, 10

        db 13, 10, 10, '        <    Balance Inquiry                   Recent Transaction    >', 13, 10, 10

        db 13, 10, 10, '        <    Transfer                      Credit Card Operations    >', 13, 10, 10

        db 13, 10, 10, '        <    Change PIN                                   Deposit    >', 13, 10, 10, 10, 10, '$'

.CODE

START:

    mov ax, @data

    mov ds, ax

    lea dx, szTitle

    mov ah, 09h

    int 21h

    ; Setup for coloring the backgrounds and texts

    mov ax, 3

    int 10h     ;

    mov ah, 06h ;

    xor al, al  ;

    xor cx, cx  ;

    ; Blue background

    mov ch, 1   ; Row start       x1

    mov cl, 0   ; Column start    y1

    mov dh, 22  ; Row end         x2

    mov dl, 80  ; Column end      y2

    mov bh, 17h ; Color

    int 10h;

    ; Header Cyan background

    mov ch, 1   ; Row start

    mov cl, 0   ; Column start

    mov dh, 3   ; Row end

    mov dl, 80  ; Column end

    mov bh, 31h ; Column end

    int 10h

    ; Log Out White background

    mov ch, 0   ; Row start

    mov cl, 0   ; Column start

    mov dh, 0   ; Row end

    mov dl, 80  ; Column end

    mov bh, 71h ; Column end

    int 10h

    ; (Withdrawal) User Input Box background

    mov ch, 6   ; Row start

    mov cl, 10  ; Column start

    mov dh, 6   ; Row end

    mov dl, 32  ; Column end

    mov bh, 31h ; Color

    int 10h

    ; Arrow

    mov ch, 6   ; Row start

    mov cl, 7  ; Column start

    mov dh, 6   ; Row end

    mov dl, 9  ; Column end

    mov bh, 71h ; Color

    int 10h

    ; (Payment) User Input Box background

    mov ch, 6   ; Row start

    mov cl, 41  ; Column start

    mov dh, 6   ; Row end

    mov dl, 67  ; Column end

    mov bh, 31h ; Color

    int 10h

    ; Arrow

    mov ch, 6   ; Row start

    mov cl, 68  ; Column start

    mov dh, 6   ; Row end

    mov dl, 70  ; Column end

    mov bh, 71h ; Color

    int 10h

    ; (Balance Inq) User Input Box background

    mov ch, 10  ; Row start

    mov cl, 10  ; Column start

    mov dh, 10  ; Row end

    mov dl, 32  ; Column end

    mov bh, 31h ; Color

    int 10h

    ; Arrow

    mov ch, 10   ; Row start

    mov cl, 7  ; Column start

    mov dh, 10  ; Row end

    mov dl, 9  ; Column end

    mov bh, 71h ; Color

    int 10h

    ; (Recent Transact) User Input Box background

    mov ch, 10  ; Row start

    mov cl, 41  ; Column start

    mov dh, 10  ; Row end

    mov dl, 67  ; Column end

    mov bh, 31h ; Color

    int 10h

    ; Arrow

    mov ch, 10   ; Row start

    mov cl, 68  ; Column start

    mov dh, 10   ; Row end

    mov dl, 70  ; Column end

    mov bh, 71h ; Color

    int 10h

    ; (Balance Inq) User Input Box background

    mov ch, 14  ; Row start

    mov cl, 10  ; Column start

    mov dh, 14  ; Row end

    mov dl, 32  ; Column end

    mov bh, 31h ; Color

    int 10h

    ; Arrow

    mov ch, 14   ; Row start

    mov cl, 7  ; Column start

    mov dh, 14  ; Row end

    mov dl, 9  ; Column end

    mov bh, 71h ; Color

    int 10h

    ; (Recent Transact) User Input Box background

    mov ch, 14  ; Row start

    mov cl, 41  ; Column start

    mov dh, 14  ; Row end

    mov dl, 67  ; Column end

    mov bh, 31h ; Color

    int 10h

    ; Arrow

    mov ch, 14   ; Row start

    mov cl, 68  ; Column start

    mov dh, 14   ; Row end

    mov dl, 70  ; Column end

    mov bh, 71h ; Color

    int 10h

    ; (Balance Inq) User Input Box background

    mov ch, 18  ; Row start

    mov cl, 10  ; Column start

    mov dh, 18  ; Row end

    mov dl, 32  ; Column end

    mov bh, 31h ; Color

    int 10h

    ; Arrow

    mov ch, 18   ; Row start

    mov cl, 7  ; Column start

    mov dh, 18  ; Row end

    mov dl, 9  ; Column end

    mov bh, 71h ; Color

    int 10h

    ; (Recent Transact) User Input Box background

    mov ch, 18  ; Row start

    mov cl, 41  ; Column start

    mov dh, 18  ; Row end

    mov dl, 67  ; Column end

    mov bh, 31h ; Color

    int 10h

    ; Arrow

    mov ch, 18   ; Row start

    mov cl, 68  ; Column start

    mov dh, 18   ; Row end

    mov dl, 70  ; Column end

    mov bh, 71h ; Color

    int 10h

    ; Yellow blinking text "Thank you"

    ;mov ch, 22  ; Row start

    ;mov cl, 2   ; Column start

    ;mov dh, 22  ; Row end

    ;mov dl, 77  ; Column end

    ;mov bh, 08EH; Blinking red background yellow text

    ;int 10h

    ; Print the Texts

    mov ax, @data

    mov ds, ax

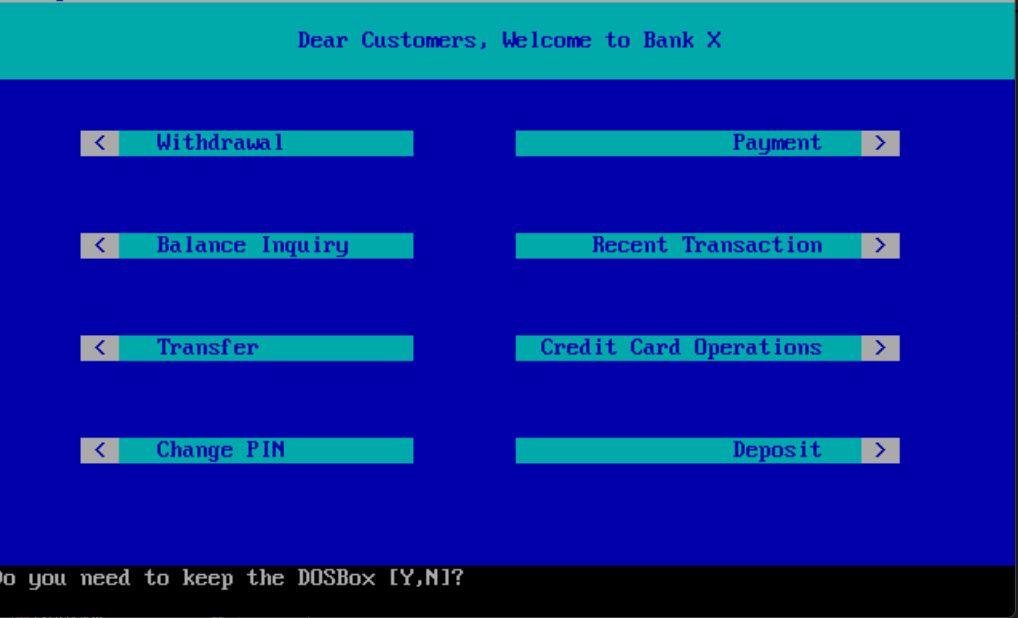
    mov ah, 09h

    lea dx, text

    int 21h

int 27h

END START



1. EXER12.ASM

; Filename: EXER12.ASM

; Programmer Name: JOHN ZILLION C. REYES

; Date: September 20, 2024

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

.MODEL small

.STACK 200h

.DATA

    szTitle db "Filename: EXER12.ASM", 0Ah, "Programmer Name: JOHN ZILLION C. REYES", 0Ah, "Date: September 20, 2024", 0Ah, "Description: This assembly language program will display multiple string variables on single lines.", 0Ah, 0Ah, '$'

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

    lea dx, szTitle

    mov ah, 09h

    int 21h

    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

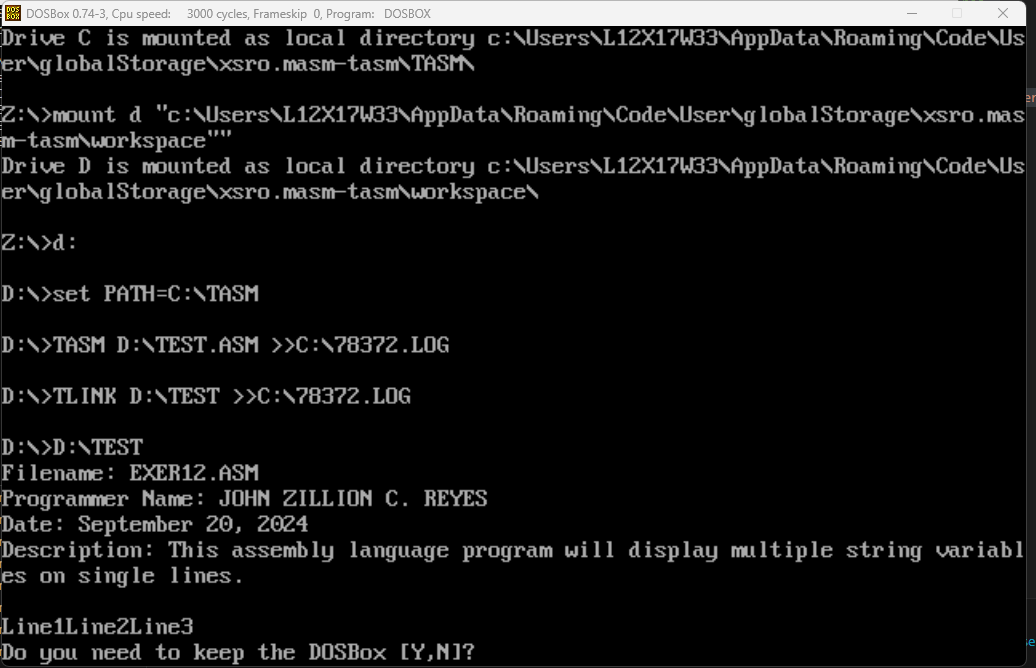
    mov dx, OFFSET String3  ; 3rd string to print

    int 21h                 ; invoke DOS to print string

    mov ah, 4ch             ; DOS terminate program function

    int 21h                 ; invoke DOS to end program

END ProgramStart



1. EXER13.ASM

; Filename: EXER13.ASM

; Programmer Name: JOHN ZILLION C. REYES

; Date: September 20, 2024

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

.MODEL small

.STACK 200h

.DATA

    szTitle db "Filename: EXER13.ASM", 0Ah, "Programmer Name: JOHN ZILLION C. REYES", 0Ah, "Date: September 20, 2024", 0Ah, "Description: This assembly language program will display multiple string variables on separate lines.", 0Ah, 0Ah, '$'

String1 DB 'Line1', 0dh, 0ah,'$'

String2 DB 'Line2', 0dh, 0ah,'$'

String3 DB 'Line3', 0dh, 0ah,'$'

; carriage return (ASCII code 0Dh) positions the cursor to the left side of the current line of characters

; line feed (ASCII code 0Ah) moves the cursor down one line on the output device.

.CODE

ProgramStart:

    mov ax, @data

    mov ds, ax

    lea dx, szTitle

    mov ah, 09h

    int 21h

    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

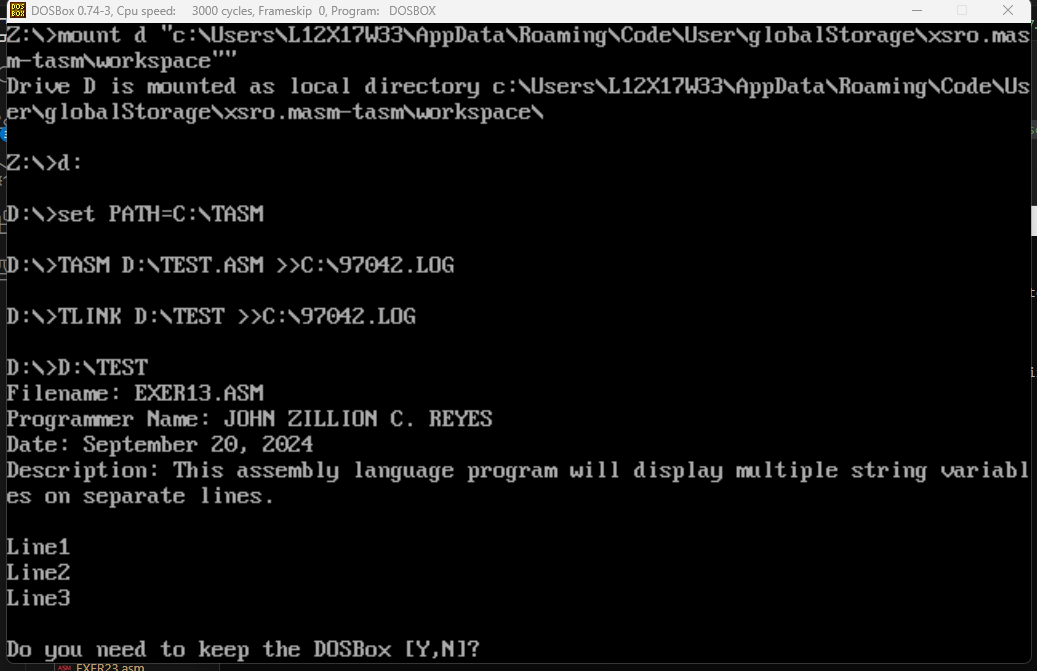
    mov dx, OFFSET String3  ; 3rd string to print

    int 21h                 ; invoke DOS to print string

    mov ah, 4ch             ; DOS terminate program function

    int 21h                 ; invoke DOS to end program

END ProgramStart



1. EXER14.ASM

; Filename: EXER14.ASM

; Programmer Name: JOHN ZILLION C. REYES

; Date: September 20, 2024

; Description: This assembly language program will get character input and

; display back character input.

.MODEL small

.STACK 200h

.DATA

    szTitle db "Filename: EXER14.ASM", 0Ah, "Programmer Name: JOHN ZILLION C. REYES", 0Ah, "Date: September 20, 2024", 0Ah, "Description: This assembly language program will get character input and ", 0Ah, "display back character input.", 0Ah, 0Ah, '$'

    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

    lea dx, szTitle

    mov ah, 09h

    int 21h

    ; 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 output message

    mov dx, offset outputMsg

    mov ah, 09h

    int 21h

    ; display the character back

    mov dl, inputChar   ; load the character to DL

    mov ah, 02h         ; function to display a character

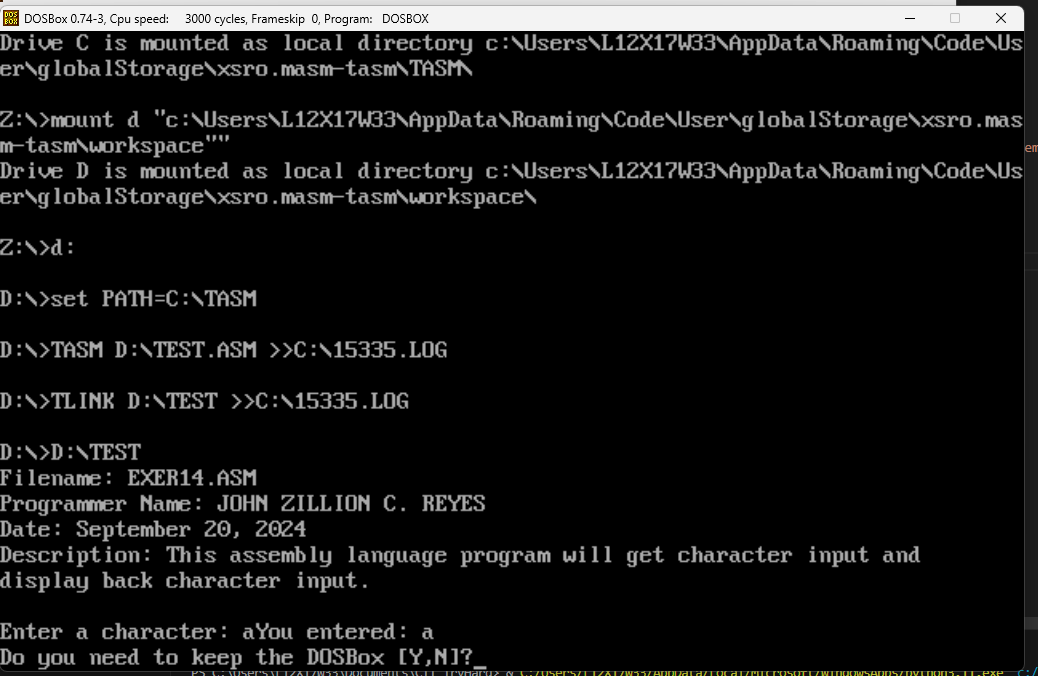
    int 21h             ; call DOS interrupt

    ; exit program

    mov ax, 4C00h       ; function to exit program

    int 21h

END Main



1. EXER15.ASM

; Filename: EXER15.ASM

; Programmer Name: JOHN ZILLION C. REYES

; Date: September 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

    szTitle db "Filename: EXER15.ASM", 0Ah, "Programmer Name: JOHN ZILLION C. REYES", 0Ah, "Date: September 20, 2024", 0Ah, "Description: This assembly language program will continuously get ", 0Ah, "character input and display back input until Enter key is pressed.", 0Ah, 0Ah, '$'

.CODE

ProgramStart:

    mov ax, @data

    mov ds, ax

    lea dx, szTitle

    mov ah, 09h

    int 21h

    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

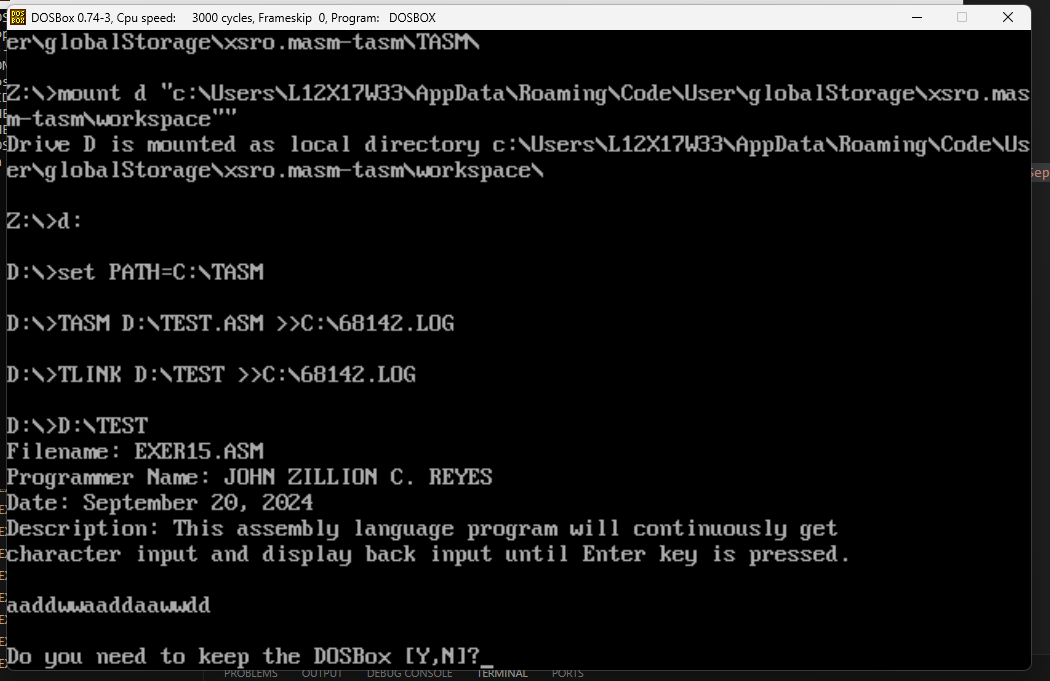
        jmp EchoLoop

    EchoDone:

        mov ah, 4ch     ; DOS terminate program function

        int 21h         ; terminate the program

END ProgramStart



1. EXER16.ASM

; Filename: EXER16.ASM

; Programmer Name: JOHN ZILLION C. REYES

; Date: September 20, 2024

; Description: This assembly language program will get character input and

; and determine if input is 'y' or 'Y' to display good morning message

; else display good afternoon message.

.MODEL small

.STACK 100h

.DATA

    szTitle db "Filename: EXER16.ASM", 0Ah, "Programmer Name: JOHN ZILLION C. REYES", 0Ah, "Date: September 20, 2024", 0Ah, "Description: This assembly language program will get character input and ", 0Ah, "and determine if input is 'y' or 'Y' to display good morning message", 0Ah, "else display good afternoon message.", 0Ah, 0Ah, '$'

    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                  ; set DS to point to data segment

    lea dx, szTitle

    mov ah, 09h

    int 21h

    mov dx, OFFSET TimePrompt   ; point to the time prompt

    mov ah, 9                   ; DOS print string function

    int 21h                     ; display time prompt

    mov ah, 1                   ; DOS get character function

    int 21h                     ; get single character response

    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

DisplayGreeting:

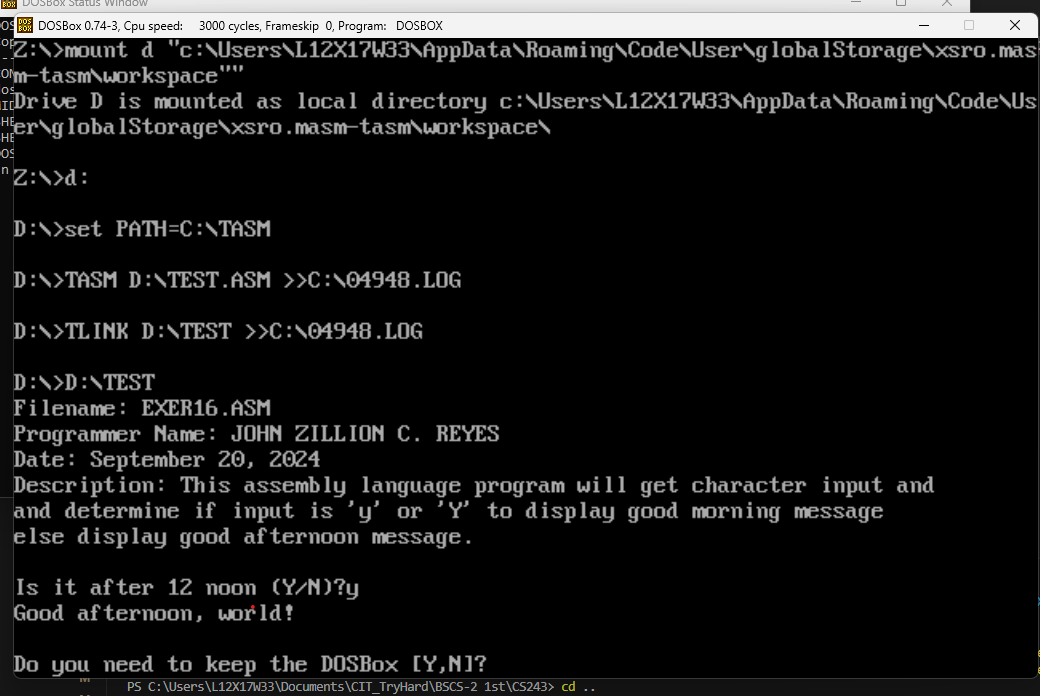
    mov ah, 9           ; DOS print string function

    int 21h             ; display the greeting

    mov ah, 4ch         ; DOS terminate program function

    int 21h             ; terminate program

END ProgramStart



1. EXER17.ASM

; Filename: EXER17.ASM

; Programmer Name: JOHN ZILLION C. REYES

; Date: September 20, 2024

; Description: This assembly language program will get string input and

; display back string.

.MODEL small

.STACK 100h

.DATA

    szTitle db "Filename: EXER17.ASM", 0Ah, "Programmer Name: JOHN ZILLION C. REYES", 0Ah, "Date: September 20, 2024", 0Ah, "Description: This assembly language program will get string input and", 0Ah, "display back string.", 0Ah, 0Ah, '$'

MAXIMUM\_STRING\_LENGTH EQU 1000

StringInput DB MAXIMUM\_STRING\_LENGTH DUP (?)

InputPrompt DB 'Enter text: $'

.CODE

ProgramStart:

    mov ax, @data

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

    lea dx, szTitle

    mov ah, 09h

    int 21h

    mov dx, OFFSET InputPrompt

    mov ah, 9

    int 21h

    mov ah, 3fh     ; DOS read 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 characters read?

    jz Done                         ; no, so you're done

    mov cx, ax                      ; put the string length in CX where

                                    ; 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

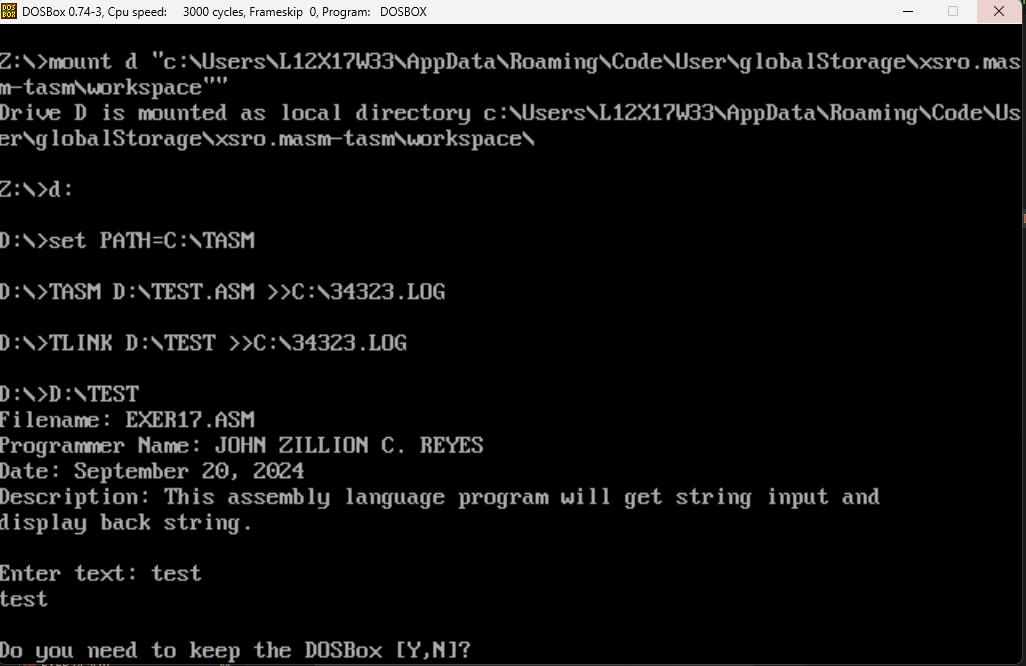
    int 21h                         ; print the string

Done:

    mov ah, 4ch

    int 21h

END ProgramStart



1. EXER18.ASM

; Filename: EXER18.ASM

; Programmer Name: JOHN ZILLION C. REYES

; Date: September 20, 2024

; Description: This assembly language program will get string input and

; display the reverse of the string.

.MODEL small

.STACK 100h

.DATA

    szTitle db "Filename: EXER18.ASM", 0Ah, "Programmer Name: JOHN ZILLION C. REYES", 0Ah, "Date: September 20, 2024", 0Ah, "Description: This assembly language program will get string input and", 0Ah, "display the reverse of the string.", 0Ah, 0Ah, '$'

MAXIMUM\_STRING\_LENGTH EQU 1000

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

    lea dx, szTitle

    mov ah, 09h

    int 21h

    mov ah, 3fh                     ;DOS read from handle function

    mov bx, 0                       ;read up to maximum number of

    mov cx, MAXIMUM\_STRING\_LENGTH   ; 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

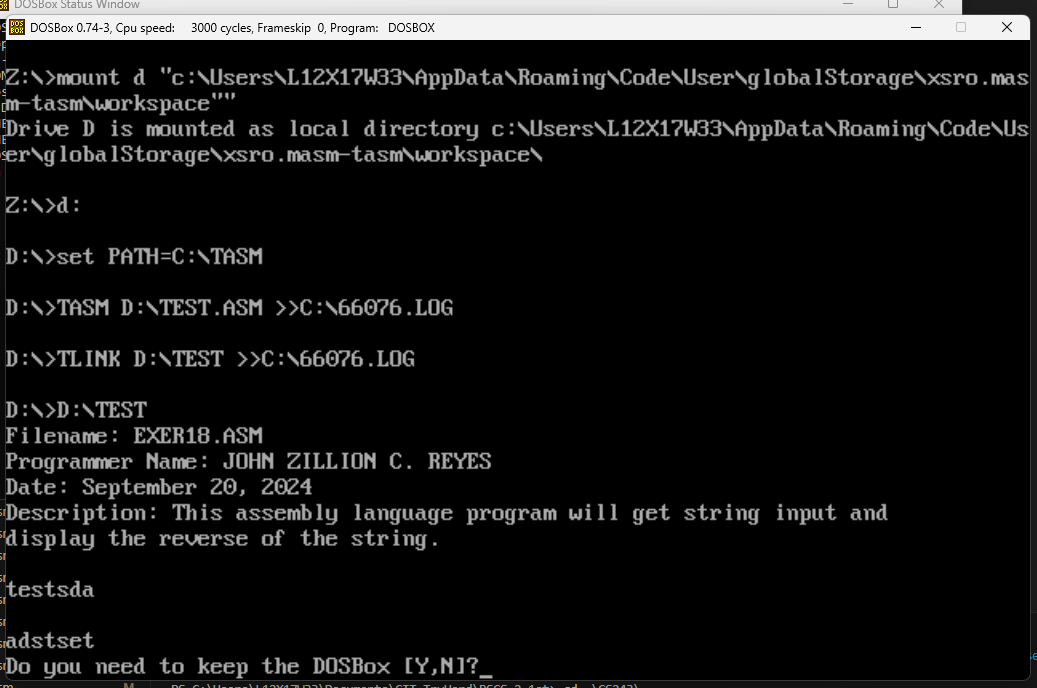
    int 21h                         ;print reversed string

Done:

    mov ah, 4ch                     ;DOS terminate program function

    int 21h                         ;terminate the program

END ProgramStart



1. EXER19.ASM

; Filename: EXER19.ASM

; Programmer Name: JOHN ZILLION C. REYES

; Date: September 20, 2024

.MODEL small

.STACK 100h

.DATA

    szTitle db "Filename: EXER19.ASM", 0Ah, "Programmer Name: JOHN ZILLION C. REYES", 0Ah, "Date: September 20, 2024", 0Ah, 0Ah, '$'

    firstIn DB 'Enter first character: $'

    secondIn DB 'Enter second character: $'

    thirdIn DB 'Enter third character: $'

    firstOut DB 'The first character is $'

    secondOut DB 'The second character is $'

    thirdOut DB 'The third character is $'

    inputIn DB 3 dup(?)

.CODE

ProgramStart:

    mov ax, @data

    mov ds, ax

    lea dx, szTitle

    mov ah, 09h

    int 21h

    mov ah, 09h

    mov dx, OFFSET firstIn

    int 21h

    mov ah, 1

    int 21h

    mov si, OFFSET inputIn

    mov [si], al

    call endLine

    mov ah, 09h

    mov dx, OFFSET secondIn

    int 21h

    mov ah, 1

    int 21h

    mov dx, OFFSET inputIn

    mov [si + 1], al

    call endLine

    mov ah, 09h

    mov dx, OFFSET thirdIn

    int 21h

    mov ah, 1

    int 21h

    mov [si + 2], al

    call endLine

    mov ah, 09h

    mov dx, OFFSET firstOut

    int 21h

    mov ah, 02h

    mov dl, [si]

    int 21h

    call endLinePeriod

    mov ah, 09h

    mov dx, OFFSET secondOut

    int 21h

    mov ah, 02h

    mov dl, [si + 1]

    int 21h

    call endLinePeriod

    mov ah, 09h

    mov dx, OFFSET thirdOut

    int 21h

    mov ah, 02h

    mov dl, [si + 2]

    int 21h

    call endLinePeriod

    mov ah, 4ch

    int 21h

endLine:

    push bx

    mov ah, 02h

    mov dl, 10

    int 21h

    pop bx

    ret

endLinePeriod:

    push bx

    mov ah, 02h

    mov dl, '.'

    int 21h

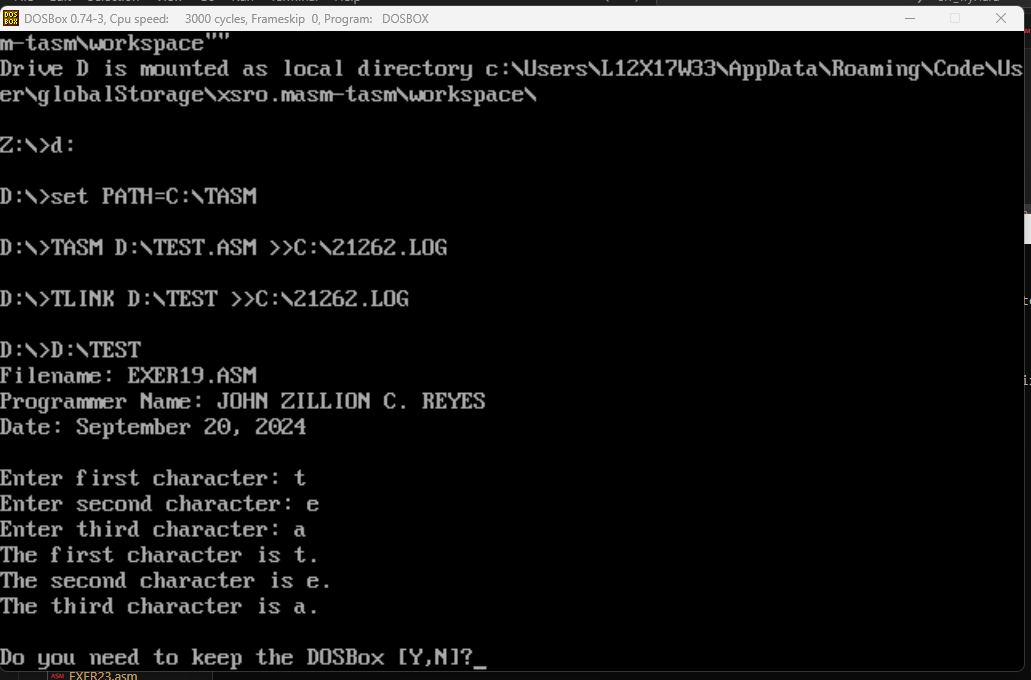
    mov dl, 10

    int 21h

    pop bx

    ret

END ProgramStart



1. EXER20.ASM

; Filename: EXER20.ASM

; Programmer Name: JOHN ZILLION C. REYES

; Date: September 20, 2024

.MODEL small

.STACK 100h

.DATA

    szTitle db "Filename: EXER20.ASM", 0Ah, "Programmer Name: JOHN ZILLION C. REYES", 0Ah, "Date: September 20, 2024", 0Ah, 0Ah, '$'

    prompt db 'Enter a character: $'

    isPrompt db 'Yes, you have entered letter A.$'

    notPrompt db 'No, you have not entered letter A. You entered character $'

.CODE

ProgramStart:

    mov ax, @data

    mov ds, ax

    lea dx, szTitle

    mov ah, 09h

    int 21h

    mov ah, 09h

    mov dx, OFFSET prompt

    int 21h

    mov ah, 1

    int 21h

    mov bl, al

    cmp al, 'A'

    mov ah, 02h

    mov dl, 0Ah

    int 21h

    je isA

    mov ah, 09h

    mov dx, OFFSET notPrompt

    int 21h

    mov ah, 02h

    mov dl, bl

    int 21h

    mov dl, '.'

    int 21h

    jmp endIsA

isA:

    mov ah, 09h

    mov dx, OFFSET isPrompt

    int 21h

endIsA:

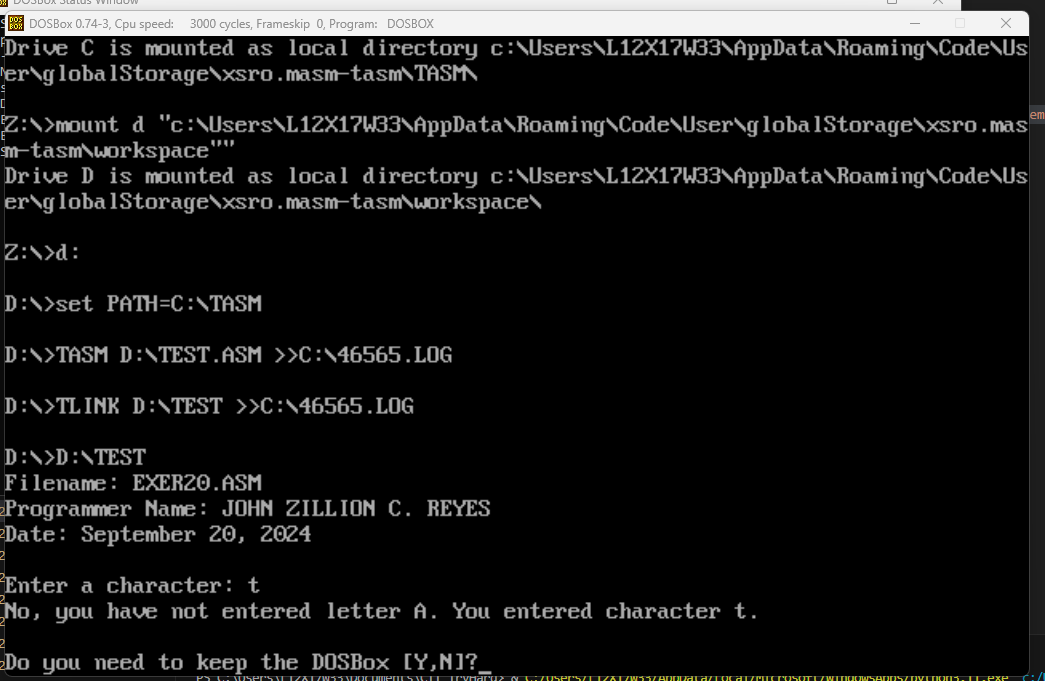
    mov ah, 02h

    mov dl, 0Ah

    int 21h

    int 27h

END ProgramStart



1. EXER21.ASM

; Filename: EXER21.ASM

; Programmer Name: JOHN ZILLION C. REYES

; Date: September 20, 2024

.MODEL small

.STACK 100h

.DATA

    szTitle db "Filename: EXER21.ASM", 0Ah, "Programmer Name: JOHN ZILLION C. REYES", 0Ah, "Date: September 20, 2024", 0Ah, 0Ah, '$'

    promptStart db 'REGISTRATION FORM', 0Ah, '$'

    promptFirst db 'Enter First Name: $'

    promptSecond db 'Enter Middle Name: $'

    promptThird db 'Enter Last Name: $'

    max\_len EQU 1000

    first db max\_len dup(?)

    middle db max\_len dup(?)

    last db max\_len dup(?)

    Msg db 'Hello, $'

.CODE

ProgramStart:

    mov ax, @data

    mov ds, ax

    lea dx, szTitle

    mov ah, 09h

    int 21h

    lea dx, promptStart

    call printString

    lea dx, promptFirst

    call printString

    lea dx, first

    mov cx, max\_len

    call getString

    push ax

    lea dx, promptSecond

    call printString

    lea dx, middle

    mov cx, max\_len

    call getString

    pop dx

    push ax

    push dx

    lea dx, promptThird

    call printString

    lea dx, last

    mov cx, max\_len

    call getString

    pop dx

    pop cx

    push ax

    push cx

    push dx

    mov dx, OFFSET Msg

    call printString

    mov dx, OFFSET first

    pop cx

    call printNumString

    mov dx, OFFSET middle

    pop cx

    call printNumString

    mov dx, OFFSET last

    pop cx

    mov ah, 40h

    add cx, -2

    mov bx, 1

    int 21h

    mov ah, 02h

    mov dl, '!'

    int 21h

    call endLine

    int 27h

getString:

    push bx

    mov ah, 3fh

    mov bx, 0

    mov cx, max\_len

    int 21h

    pop bx

    ret

printNumString:

    push ax

    push bx

    mov ah, 40h

    add cx, -2

    mov bx, 1

    int 21h

    mov ah, 02h

    mov dl, ' '

    int 21h

    pop bx

    pop ax

    ret

printString:

    push ax

    mov ah, 09h

    int 21h

    pop ax

    ret

endLine:

    push ax

    push bx

    mov ah, 02h

    mov dl, 0Ah

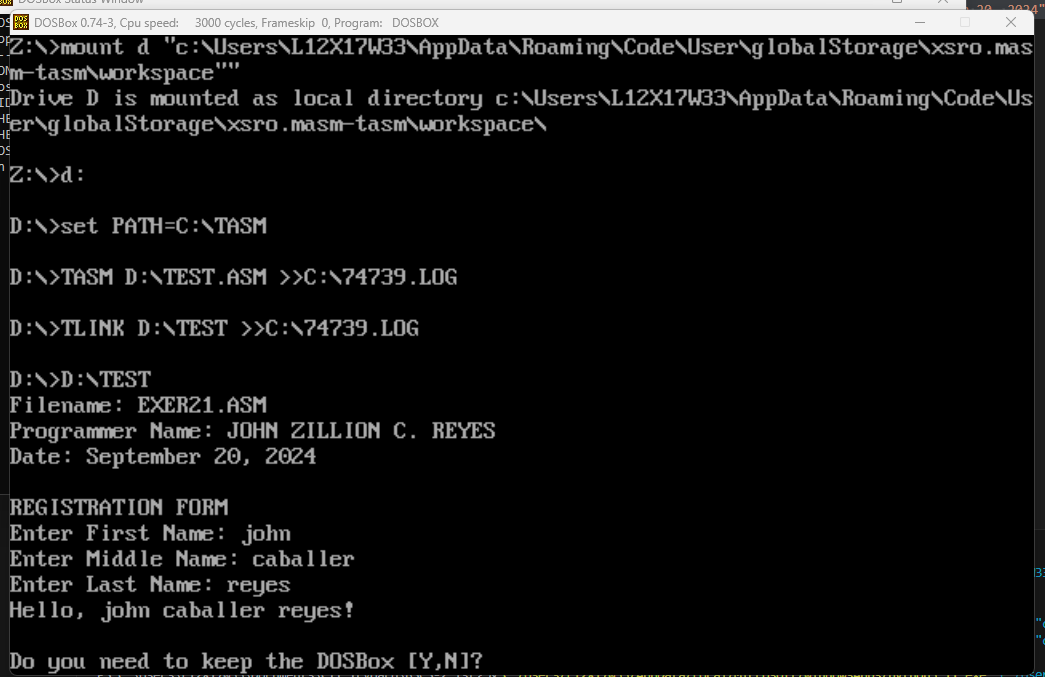
    int 21h

    pop bx

    pop ax

    ret

END ProgramStart



1. EXER22.ASM

; Filename: EXER22.ASM

; Programmer Name: JOHN ZILLION C. REYES

; Date: September 20, 2024

.MODEL small

.STACK 100h

.DATA

    szTitle db "Filename: EXER22.ASM", 0Ah, "Programmer Name: JOHN ZILLION C. REYES", 0Ah, "Date: September 20, 2024", 0Ah, 0Ah, '$'

    promptStart db 'Cebu Institute of Technology - University', 0Ah, 'STUDENT ENROLLMENT FORM', 0Ah, 0Ah, 'Please enter the following information:', 0Ah, '$'

    promptID db 'Student ID Number: ', '$'

    promptFirst db 'First Name: ', '$'

    promptMiddle db 'Middle Name: ', '$'

    promptLast db 'Last Name: ', '$'

    promptAddress db 'Address: ', '$'

    promptCourse db 'Course: ', '$'

    promptYear db 'Year: ', '$'

    promptBMon db 'Birthday Month: ', '$'

    promptBDay db 'Birthday Day: ', '$'

    promptBYear db 'Birthday Year: ', '$'

    promptEmail db 'Email Address: ', '$'

    headerOutput db 0Ah, 'SUMMARY', 0Ah, 'Please check if all information are correct.', 0Ah, 0Ah, '$'

    outID db 'ID Number: ', '$'

    outName db 'Full Name: ', '$'

    outAddress db 'Address: ', '$'

    outCourseYear db 'Course & Year: ', '$'

    outBirthday db 'Birthday: ', '$'

    outEmail db 'Email Address: ', '$'

    footer db 0Ah, 'Thank you for enrolling at CIT-U.', 0Ah, 'Copyright 2024', 0Ah, 'Programmer: JOHN ZILLION REYES', 0Ah, '$'

    max\_len EQU 1002

    inId db max\_len dup(?)

    inFirst db max\_len dup(?)

    inMiddle db max\_len dup(?)

    inLast db max\_len dup(?)

    inAddress db max\_len dup(?)

    inCourse db max\_len dup(?)

    inYear db max\_len dup(?)

    inBMon db max\_len dup(?)

    inBDay db max\_len dup(?)

    inBYear db max\_len dup(?)

    inEmail db max\_len dup(?)

.CODE

ProgramStart:

    mov ax, @data

    mov ds, ax

    lea dx, szTitle

    mov ah, 09h

    int 21h

    lea dx, promptStart

    call printString

    lea bx, promptID

    lea dx, inId

    call inputField

    lea bx, promptFirst

    lea dx, inFirst

    call inputField

    lea bx, promptMiddle

    lea dx, inMiddle

    call inputField

    lea bx, promptLast

    lea dx, inLast

    call inputField

    lea bx, promptAddress

    lea dx, inAddress

    call inputField

    lea bx, promptCourse

    lea dx, inCourse

    call inputField

    lea bx, promptYear

    lea dx, inYear

    call inputField

    lea bx, promptBMon

    lea dx, inBMon

    call inputField

    lea bx, promptBDay

    lea dx, inBDay

    call inputField

    lea bx, promptBYear

    lea dx, inBYear

    call inputField

    lea bx, promptEmail

    lea dx, inEmail

    call inputField

    lea dx, headerOutput

    call printString

    lea bx, outID

    lea dx, inId

    call outputField

    lea dx, outName

    call printString

    lea dx, inLast

    call printNumString

    call addComma

    call addSpace

    lea dx, inFirst

    call printNumString

    call addSpace

    lea dx, inMiddle

    call printNumString

    call endLine

    lea bx, outAddress

    lea dx, inAddress

    call outputField

    lea dx, outCourseYear

    call printString

    lea dx, inCourse

    call printNumString

    call addSpace

    lea dx, inYear

    call printNumString

    call endLine

    lea dx, outBirthday

    call printString

    lea dx, inBMon

    call printNumString

    call addSpace

    lea dx, inBDay

    call printNumString

    call addComma

    call addSpace

    lea dx, inBYear

    call printNumString

    call endLine

    lea bx, outEmail

    lea dx, inEmail

    call outputField

    lea dx, footer

    call printString

    int 27h

outputField:

    push dx

    mov dx, bx

    call printString

    pop dx

    call printNumString

    call endLine

    ret

inputField:

    push dx

    mov dx, bx

    call printString

    pop dx

    call getString

    ret

getString:

    push ax

    push bx

    push cx

    add dx, 2

    mov ah, 3fh

    mov bx, 0

    mov cx, max\_len - 2

    int 21h

    sub dx, 2

    mov si, dx

    mov [si], ah

    mov [si + 1], al

    pop cx

    pop bx

    pop ax

    ret

printNumString:

    push ax

    push bx

    push cx

    mov si, dx

    mov ah, 40h

    mov ch, [si]

    mov cl, [si + 1]

    add dx, 2

    sub cx, 2

    mov bx, 1

    int 21h

    pop cx

    pop bx

    pop ax

    ret

printString:

    push ax

    mov ah, 09h

    int 21h

    pop ax

    ret

endLine:

    push ax

    push bx

    mov ah, 02h

    mov dl, 0Ah

    int 21h

    pop bx

    pop ax

    ret

addSpace:

    push ax

    push bx

    mov ah, 02h

    mov dl, ' '

    int 21h

    pop bx

    pop ax

    ret

addComma:

    push ax

    push bx

    mov ah, 02h

    mov dl, ','

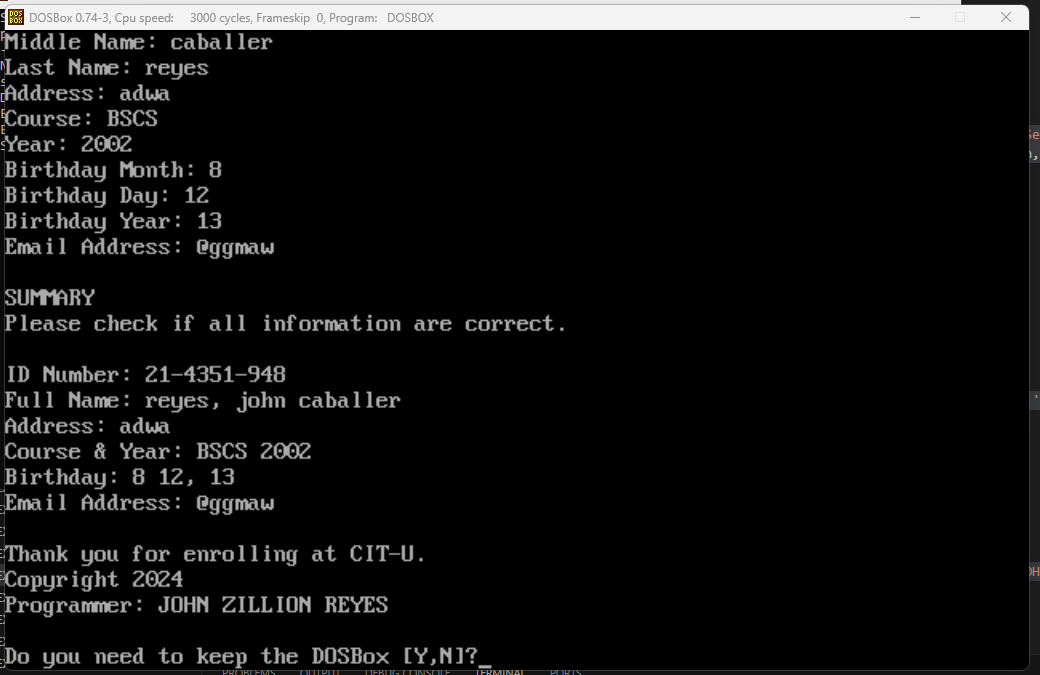
    int 21h

    pop bx

    pop ax

    ret

END ProgramStart



1. EXER23.ASM

; Filename: EXER23.ASM

; Programmer Name: JOHN ZILLION C. REYES

; Date: September 20, 2024

.MODEL small

.STACK 100h

.DATA

    szTitle db "Filename: EXER23.ASM", 0Ah, "Programmer Name: JOHN ZILLION C. REYES", 0Ah, "Date: September 20, 2024", 0Ah, 0Ah, '$'

    header1 db 'Cebu Institute of Technology - University', 0Ah, 'Automated Teller Machine', 0Ah, 0Ah, 'MAIN MENU', 0Ah, '$'

    header2 db '1 Balance Inquiry', 0Ah, '2 Withdrawal', 0Ah, '3 Deposit', 0Ah, '4 Transfer', 0Ah, '5 Bills Payment', 0Ah, '6 Change Pin', 0Ah, '7 Exit', 0Ah, 'Enter number of your choice: $'

    opStart db 0Ah, 0Ah, 'You have chosen ' , '$'

    op1 db '1 Balance Inquiry.$'

    op2 db '2 Withdrawal.$'

    op3 db '3 Deposit.$'

    op4 db '4 Transfer.$'

    op5 db '5 Bills Payment.$'

    op6 db '6 Change Pin.$'

    op7 db '7 Exit.$'

    footer db 0Ah, 0Ah, 'Thank you for banking with us!', 0Ah, 'Copyright 2024', 0Ah, 'Programmer: JOHN ZILLION REYES$'

.CODE

ProgramStart:

    mov ax, @data

    mov ds, ax

    lea dx, szTitle

    mov ah, 09h

    int 21h

    lea dx, header1

    call printString

    lea dx, header2

    call printString

    mov ah, 01h

    int 21h

    lea dx, opStart

    call printString

    cmp al, '1'

    je oneOp

    cmp al, '2'

    je twoOp

    cmp al, '3'

    je threeOp

    cmp al, '4'

    je fourOp

    cmp al, '5'

    je fiveOp

    cmp al, '6'

    je sixOp

    cmp al, '7'

    je sevenOp

    jmp exitOp

oneOp:

    lea dx, op1

    call printString

    jmp exitOp

twoOp:

    lea dx, op2

    call printString

    jmp exitOp

threeOp:

    lea dx, op3

    call printString

    jmp exitOp

fourOp:

    lea dx, op4

    call printString

    jmp exitOp

fiveOp:

    lea dx, op5

    call printString

    jmp exitOp

sixOp:

    lea dx, op6

    call printString

    jmp exitOp

sevenOp:

    lea dx, op7

    call printString

    jmp exitOp

exitOp:

    lea dx, footer

    call printString

    int 27h

printString:

    push ax

    mov ah, 09h

    int 21h

    pop ax

    ret

endLine:

    push ax

    push bx

    mov ah, 02h

    mov dl, 0Ah

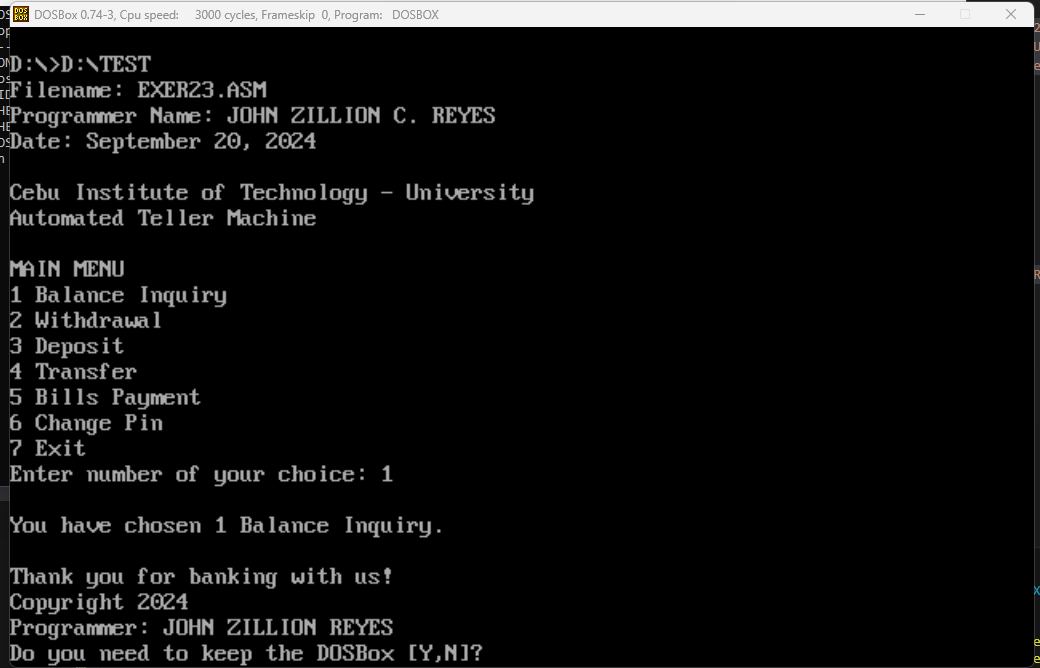
    int 21h

    pop bx

    pop ax

    ret

END ProgramStart



1. EXER24.ASM

; Filename: EXER24.ASM

; Programmer Name: JOHN ZILLION C. REYES

; Date: September 20, 2024

.model small

.stack 100

.data

    szTitle db "Filename: EXER24.ASM", 0Ah, "Programmer Name: JOHN ZILLION C. REYES", 0Ah, "Date: September 20, 2024", 0Ah, 0Ah, '$'

    header db 'Cebu Institute of Technology - University', 0Ah, 'VEHICLE STICKER APPLICATION FORM', 0Ah, 'Please enter the needed information:', 0Ah, 0Ah, '$'

    max\_len EQU 1000

    in1 db max\_len dup(' ')

    in2 db max\_len dup(' ')

    in3 db max\_len dup(' ')

    in4 db max\_len dup(' ')

    in5 db max\_len dup(' ')

    in6 db max\_len dup(' ')

    in7 db max\_len dup(' ')

    in8 db max\_len dup(' ')

    in9 db max\_len dup(' ')

    in10 db max\_len dup(' ')

    h0 db 0Ah, 'SUMMARY', 0Ah, '$'

    h1 db "Cebu Institute of Technology - University$"

    h2 db "VEHICLE STICKER APPLICATION FORM$"

    h3 db "Please fill out form below$"

    d1 db "Personnel Type:$"

    d2 db "Name of Applicant/Driver:$"

    d3 db "Mobile Number:$"

    d4 db "Vehicle Make(s)/Brand:$"

    d5 db "Vehicle Color:$"

    d6 db "Vehicle Sticker Type:$"

    d7 db "ID Number:$"

    d8 db "Address:$"

    d9 db "Plate Number:$"

    d10 db "Vehicle Type:$"

    b1 db "SUBMIT$"

    f1 db "Copyright 2024 JOHN ZILLION C. REYES$"

    f2 db "Thank you!$"

.code

start:

    mov ax, @data

    mov ds, ax

    lea dx, szTitle

    mov ah, 09h

    int 21h

    lea dx, header

    call printString

    lea bx, d1

    lea dx, in1

    call inputData

    lea bx, d2

    lea dx, in2

    call inputData

    lea bx, d3

    lea dx, in3

    call inputData

    lea bx, d4

    lea dx, in4

    call inputData

    lea bx, d5

    lea dx, in5

    call inputData

    lea bx, d6

    lea dx, in6

    call inputData

    lea bx, d7

    lea dx, in7

    call inputData

    lea bx, d8

    lea dx, in8

    call inputData

    lea bx, d9

    lea dx, in9

    call inputData

    lea bx, d10

    lea dx, in10

    call inputData

    lea dx, h0

    call printString

    call endLine

    call printMargin

    mov cx, 73

    mov bl, 4Fh

    call printColor

    mov cx, 16

    call printSpace

    lea dx, h1

    call printString

    mov cx,16

    call printSpace

    call printMargin

    call endLine

    call printMargin

    mov cx, 73

    mov bl, 4Fh

    call printColor

    mov cx, 20

    call printSpace

    lea dx, h2

    call printString

    mov cx, 21

    call printSpace

    call printMargin

    call endLine

    call printMargin

    mov cx, 73

    mov bl, 0CEh

    call printColor

    mov cx, 23

    call printSpace

    lea dx, h3

    call printString

    mov cx, 24

    call printSpace

    call printMargin

    call endLine

    call addSpacing

    call printMargin

    mov bl, 70h

    mov cx, 25

    call printColor

    lea dx, d1

    call printString

    mov cx, 10

    call printSpace

    mov al, 1

    lea dx, in1

    call printBlank

    call printMargin

    mov cx, 26

    call printColor

    lea dx, d6

    call printString

    mov cx, 5

    call printSpace

    lea dx, in6

    call printBlank

    call printMargin

    call endLine

    call addSpacing

    call printMargin

    mov bl, 70h

    mov cx, 25

    call printColor

    lea dx, d2

    call printString

    mov al, 0

    lea dx, in2

    call printBlank

    call printMargin

    mov cx, 26

    call printColor

    lea dx, d7

    call printString

    mov cx, 16

    call printSpace

    lea dx, in7

    call printBlank

    call printMargin

    call endLine

    call addSpacing

    call printMargin

    mov bl, 70h

    mov cx, 25

    call printColor

    lea dx, d3

    call printString

    mov cx, 11

    call printSpace

    mov al, 0

    lea dx, in3

    call printBlank

    call printMargin

    mov cx, 26

    call printColor

    lea dx, d8

    call printString

    mov cx, 18

    call printSpace

    lea dx, in8

    call printBlank

    call printMargin

    call endLine

    call addSpacing

    call printMargin

    mov bl, 70h

    mov cx, 25

    call printColor

    lea dx, d4

    call printString

    mov cx, 3

    call printSpace

    mov al, 0

    lea dx, in4

    call printBlank

    call printMargin

    mov cx, 26

    call printColor

    lea dx, d9

    call printString

    mov cx, 13

    call printSpace

    lea dx, in9

    call printBlank

    call printMargin

    call endLine

    call addSpacing

    call printMargin

    mov bl, 70h

    mov cx, 25

    call printColor

    lea dx, d5

    call printString

    mov cx, 11

    call printSpace

    lea dx, in5

    mov al, 1

    call printBlank

    call printMargin

    mov cx, 26

    call printColor

    lea dx, d10

    call printString

    mov cx, 13

    call printSpace

    lea dx, in10

    call printBlank

    call printMargin

    call endLine

    call addSpacing

    call addSpacing

    call printMargin

    mov bl, 70h

    mov cx, 31

    call printColor

    call printSpace

    mov bl, 4Eh

    mov cx, 10

    call printColor

    mov cx, 2

    call printSpace

    lea dx, b1

    call printString

    call printSpace

    mov bl, 70h

    mov cx, 32

    call printColor

    call printSpace

    call printMargin

    call endLine

    call addSpacing

    mov bl, 70h

    mov cx, 75

    call printColor

    mov cx, 2

    call printSpace

    mov cx, 18

    call printSpace

    lea dx, f1

    call printString

    mov cx, 19

    call printSpace

    call printMargin

    call endLine

    call addSpacing

    call endLine

    mov bl, 8Eh

    mov cx, 80

    call printColor

    mov cx, 33

    call printSpace

    lea dx, f2

    call printString

    call endLine

    int 27h

endLine:

    push ax

    push dx

    push cx

    mov ah, 02h

    mov dl, 0Ah

    int 21h

    mov cx, 1

    call printSpace

    pop cx

    pop dx

    pop ax

    ret

printBlank:

    push bx

    push cx

    push ax

    mov bl, 0Fh

    cmp al, 1

    je withBlank

    mov cx, 10

    call printColor

    call outputData

    jmp endBlank

withBlank:

    mov cx, 7

    call printColor

    call outputData

    mov cx, 3

    mov bl, 4Fh

    call printColor

    mov cx, 1

    call printSpace

    mov ah, 02h

    mov dl, 'V'

    int 21h

    call printSpace

endBlank:

    pop ax

    pop cx

    pop bx

    ret

printMargin:

    push bx

    push cx

    mov bl, 70h

    mov cx, 2

    call printColor

    call printSpace

    pop cx

    pop bx

    ret

printString:

    push ax

    mov ah, 09h

    int 21h

    pop ax

    ret

printColor:

    push ax

    mov ah, 09h

    mov al, 0

    int 10h

    pop ax

    ret

printSpace:

    push ax

    push dx

    push cx

printSpaceLoop:

    mov ah, 02h

    mov dl, ' '

    int 21h

    loop printSpaceLoop

    pop cx

    pop dx

    pop ax

    ret

addSpacing:

    push bx

    push cx

    call printMargin

    mov bl, 70h

    mov cx, 73

    call printColor

    call printSpace

    call printMargin

    call endLine

    pop cx

    pop bx

    ret

inputData:

    push cx

    push dx

    mov dx, bx

    call printString

    mov cx, 1

    call printSpace

    pop dx

    mov ah, 3Fh

    mov bx, 0

    mov cx, max\_len

    int 21h

    mov si, dx

    add si, ax

    mov [si - 2], ' '

    mov [si - 1], ' '

    pop cx

    ret

outputData:

    push ax

    push bx

    mov ah, 40h

    mov bx, 1

    int 21h

    pop bx

    pop ax

    ret

end start



1. EXER25.ASM

; Filename: EXER25.ASM

; Programmer Name: JOHN ZILLION C. REYES

; Date: September 29, 2024

; Description: This assembly language program will get 3 string inputs and

; display back the 3 strings on separate lines.

.MODEL SMALL

.STACK 100h

.DATA

    szTitle db "Filename: EXER25.ASM", 0Ah, "Programmer Name: JOHN ZILLION C. REYES", 0Ah, "Date: September 29, 2024", 0Ah, "Description: This assembly language program will get 3 string inputs and", 0Ah, "display back the 3 strings on separate lines.", 0Ah, 0Ah, '$'

    inputString1 DB 50 DUP('$') ; Reserve 50 bytes for input string

    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

    lea dx, szTitle

    mov ah, 09h

    int 21h

    ; 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 ; DOS function to display a string

    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 ; DOS function to display a string

    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 (length and max length)

    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 (length and max length)

    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 (length and max length)

    MOV AH, 09h ; DOS function to display a string

    INT 21h

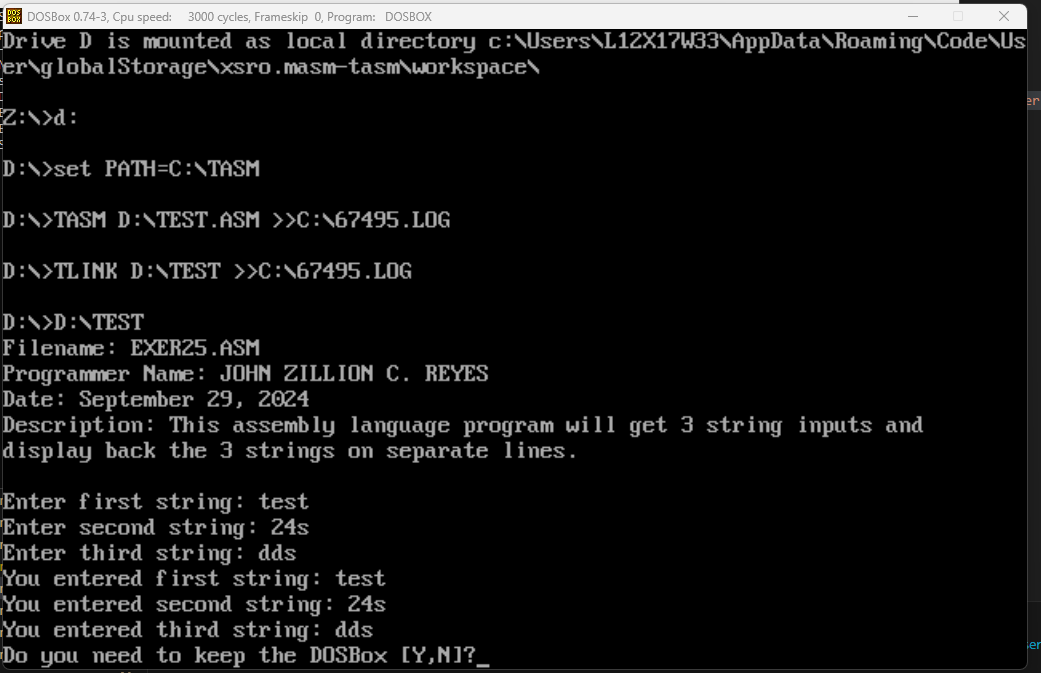
    ; Exit program

    MOV AX, 4C00h ; DOS function to terminate program

    INT 21h

    MAIN ENDP

    END MAIN



1. EXER26.ASM

; Filename: EXER26.ASM

; Programmer Name: JOHN ZILLION C. REYES

; Date: September 29, 2024

.model small

.stack 100

.data

    szTitle db "Filename: EXER26.ASM", 0Ah, "Programmer Name: JOHN ZILLION C. REYES", 0Ah, "Date: September 29, 2024", 0Ah, 0Ah, '$'

    h1 db "Input first number: $"

    h2 db "Input second number: $"

    Op db " + $"

    Res db " = $"

    cls db "                             ", 0Dh , "                             ", 0Dh, "$"

.code

start:

    mov ax, @data

    mov ds, ax

    lea dx, szTitle

    mov ah, 09h

    int 21h

    lea di, h1

    call inputNum

    mov bx, ax

    lea di, h2

    call inputNum

    mov cx, ax

    call endLine

    mov ax, bx

    call printNum

    lea dx, Op

    call printString

    mov ax, cx

    call printNum

    add ax, bx

    lea dx, Res

    call printString

    call printNum

    call endLine

    int 27h

isOdd:

    push bx

    push dx

    mov dx, 0

    mov bx, 2

    div bx

    cmp dx, 1

    pop dx

    pop bx

    ret

inputNum:

    push bx

    push cx

    push dx

    push si

    mov cx, 0

    mov si, 10

    mov bx, 0

    mov dx, di

    call printString

inputLoop:

    mov ah, 7

    int 21h

    cmp al, 8

    je inputRem

    cmp al, '0'

    jl exitInputNum

    cmp al, '9'

    jg exitInputNum

    sub al, '0'

    mov cl, al

    mov ax, bx

    mul si

    add ax, cx

    mov bx, ax

    mov dx, offset cls

    call printString

    mov dx, di

    call printString

    mov ax, bx

    call printNum

    jmp inputLoop

inputRem:

    mov ax, bx

    mov dx, 0

    div si

    mov bx, ax

    mov dx, offset cls

    call printString

    mov dx, di

    call printString

    mov ax, bx

    call printNum

    jmp inputLoop

exitInputNum:

    call endLine

    mov ax, bx

    pop si

    pop dx

    pop cx

    pop bx

    ret

printNum:

    push ax

    push bx

    push cx

    push dx

    push si

    mov cx, 0

digitLoop:

    mov bx, 10

    mov dx, 0

    div bx

    mov bx, ax

    mov ax, cx

    call isOdd

    je ifAppend

    mov ah, dl

    mov al, 0

    jmp endifAppend

ifAppend:

    pop ax

    mov al, dl

endifAppend:

    push ax

    inc cx

    mov ax, bx

    cmp ax, 0

    jne digitLoop

printLoop:

    mov ax, cx

    call isOdd

    pop ax

    je ifPrint

    mov dl, al

    push ax

    jmp endifPrint

ifPrint:

    mov dl, ah

endifPrint:

    add dl, '0'

    mov ah, 02h

    int 21h

    loop printLoop

    pop si

    pop dx

    pop cx

    pop bx

    pop ax

    ret

printString:

    push ax

    mov ah, 09h

    int 21h

    pop ax

    ret

endLine:

    push ax

    push dx

    mov ah, 02h

    mov dl, 0Ah

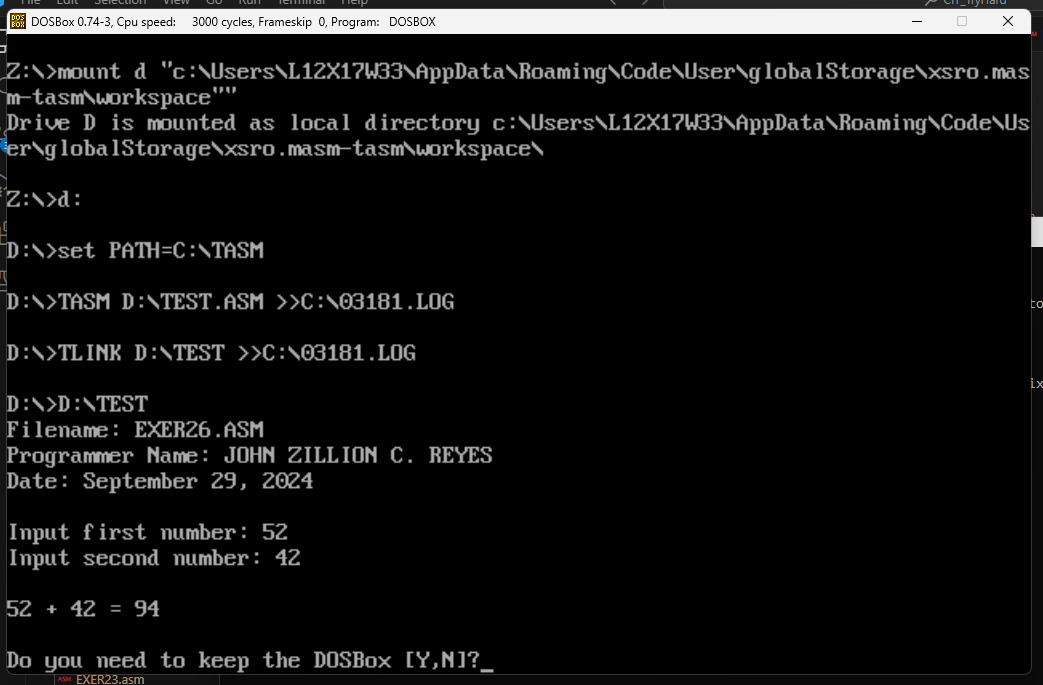
    int 21h

    pop dx

    pop ax

    ret

end start



1. EXER27.ASM

; Filename: EXER27.ASM

; Programmer Name: JOHN ZILLION C. REYES

; Date: September 29, 2024

.model small

.stack 100

.data

    szTitle db "Filename: EXER27.ASM", 0Ah, "Programmer Name: JOHN ZILLION C. REYES", 0Ah, "Date: September 29, 2024", 0Ah, 0Ah, '$'

    h1 db "Input first number: $"

    h2 db "Input second number: $"

    Op db " - $"

    Res db " = $"

    cls db "                             ", 0Dh , "                             ", 0Dh, "$"

.code

start:

    mov ax, @data

    mov ds, ax

    lea dx, szTitle

    mov ah, 09h

    int 21h

    lea di, h1

    call inputNum

    mov bx, ax

    lea di, h2

    call inputNum

    mov cx, ax

    call endLine

    mov ax, bx

    call printNum

    lea dx, Op

    call printString

    mov ax, cx

    call printNum

    sub bx, ax

    lea dx, Res

    call printString

    mov ax, bx

    call printNum

    call endLine

    int 27h

isOdd:

    push bx

    push dx

    mov dx, 0

    mov bx, 2

    div bx

    cmp dx, 1

    pop dx

    pop bx

    ret

inputNum:

    push bx

    push cx

    push dx

    push si

    mov cx, 0

    mov si, 10

    mov bx, 0

    mov dx, di

    call printString

inputLoop:

    mov ah, 7

    int 21h

    cmp al, 8

    je inputRem

    cmp al, '0'

    jl exitInputNum

    cmp al, '9'

    jg exitInputNum

    sub al, '0'

    mov cl, al

    mov ax, bx

    mul si

    add ax, cx

    mov bx, ax

    mov dx, offset cls

    call printString

    mov dx, di

    call printString

    mov ax, bx

    call printNum

    jmp inputLoop

inputRem:

    mov ax, bx

    mov dx, 0

    div si

    mov bx, ax

    mov dx, offset cls

    call printString

    mov dx, di

    call printString

    mov ax, bx

    call printNum

    jmp inputLoop

exitInputNum:

    call endLine

    mov ax, bx

    pop si

    pop dx

    pop cx

    pop bx

    ret

printNum:

    push ax

    push bx

    push cx

    push dx

    push si

    mov cx, 0

digitLoop:

    mov bx, 10

    mov dx, 0

    div bx

    mov bx, ax

    mov ax, cx

    call isOdd

    je ifAppend

    mov ah, dl

    mov al, 0

    jmp endifAppend

ifAppend:

    pop ax

    mov al, dl

endifAppend:

    push ax

    inc cx

    mov ax, bx

    cmp ax, 0

    jne digitLoop

printLoop:

    mov ax, cx

    call isOdd

    pop ax

    je ifPrint

    mov dl, al

    push ax

    jmp endifPrint

ifPrint:

    mov dl, ah

endifPrint:

    add dl, '0'

    mov ah, 02h

    int 21h

    loop printLoop

    pop si

    pop dx

    pop cx

    pop bx

    pop ax

    ret

printString:

    push ax

    mov ah, 09h

    int 21h

    pop ax

    ret

endLine:

    push ax

    push dx

    mov ah, 02h

    mov dl, 0Ah

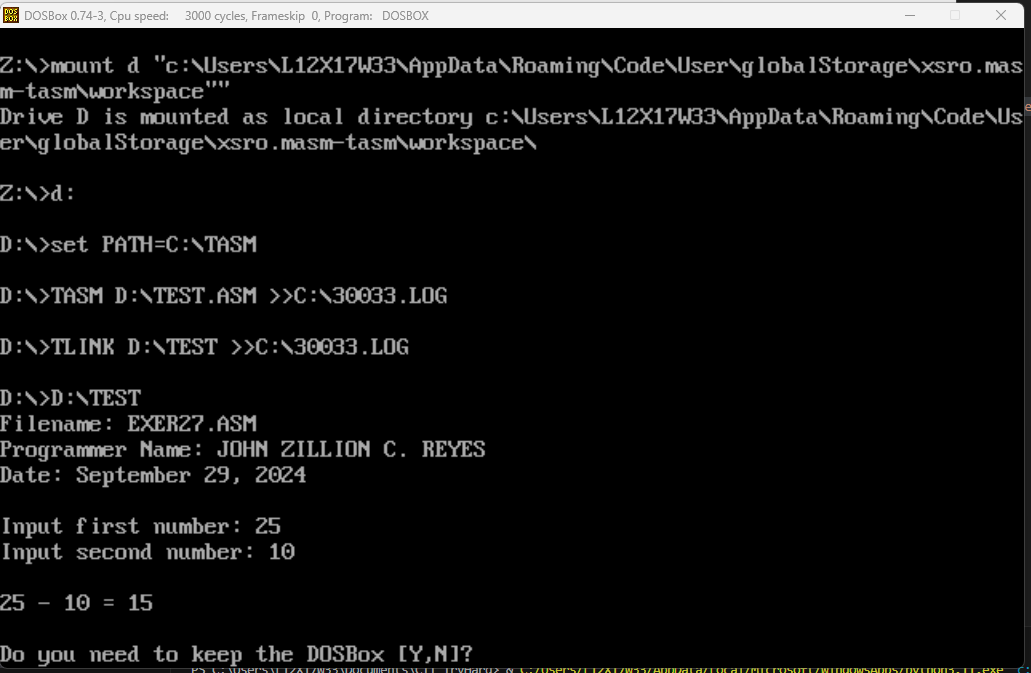
    int 21h

    pop dx

    pop ax

    ret

end start



1. EXER28.ASM

; Filename: EXER28.ASM

; Programmer Name: JOHN ZILLION C. REYES

; Date: September 29, 2024

.model small

.stack 100

.data

    szTitle db "Filename: EXER28.ASM", 0Ah, "Programmer Name: JOHN ZILLION C. REYES", 0Ah, "Date: September 29, 2024", 0Ah, 0Ah, '$'

    h1 db "Input first number: $"

    h2 db "Input second number: $"

    Op db " x $"

    Res db " = $"

    cls db "                             ", 0Dh , "                             ", 0Dh, "$"

.code

start:

    mov ax, @data

    mov ds, ax

    lea dx, szTitle

    mov ah, 09h

    int 21h

    lea di, h1

    call inputNum

    mov bx, ax

    lea di, h2

    call inputNum

    mov cx, ax

    call endLine

    mov ax, bx

    call printNum

    lea dx, Op

    call printString

    mov ax, cx

    call printNum

    mul bx

    lea dx, Res

    call printString

    call printNum

    call endLine

    int 27h

isOdd:

    push bx

    push dx

    mov dx, 0

    mov bx, 2

    div bx

    cmp dx, 1

    pop dx

    pop bx

    ret

inputNum:

    push bx

    push cx

    push dx

    push si

    mov cx, 0

    mov si, 10

    mov bx, 0

    mov dx, di

    call printString

inputLoop:

    mov ah, 7

    int 21h

    cmp al, 8

    je inputRem

    cmp al, '0'

    jl exitInputNum

    cmp al, '9'

    jg exitInputNum

    sub al, '0'

    mov cl, al

    mov ax, bx

    mul si

    add ax, cx

    mov bx, ax

    mov dx, offset cls

    call printString

    mov dx, di

    call printString

    mov ax, bx

    call printNum

    jmp inputLoop

inputRem:

    mov ax, bx

    mov dx, 0

    div si

    mov bx, ax

    mov dx, offset cls

    call printString

    mov dx, di

    call printString

    mov ax, bx

    call printNum

    jmp inputLoop

exitInputNum:

    call endLine

    mov ax, bx

    pop si

    pop dx

    pop cx

    pop bx

    ret

printNum:

    push ax

    push bx

    push cx

    push dx

    push si

    mov cx, 0

digitLoop:

    mov bx, 10

    mov dx, 0

    div bx

    mov bx, ax

    mov ax, cx

    call isOdd

    je ifAppend

    mov ah, dl

    mov al, 0

    jmp endifAppend

ifAppend:

    pop ax

    mov al, dl

endifAppend:

    push ax

    inc cx

    mov ax, bx

    cmp ax, 0

    jne digitLoop

printLoop:

    mov ax, cx

    call isOdd

    pop ax

    je ifPrint

    mov dl, al

    push ax

    jmp endifPrint

ifPrint:

    mov dl, ah

endifPrint:

    add dl, '0'

    mov ah, 02h

    int 21h

    loop printLoop

    pop si

    pop dx

    pop cx

    pop bx

    pop ax

    ret

printString:

    push ax

    mov ah, 09h

    int 21h

    pop ax

    ret

endLine:

    push ax

    push dx

    mov ah, 02h

    mov dl, 0Ah

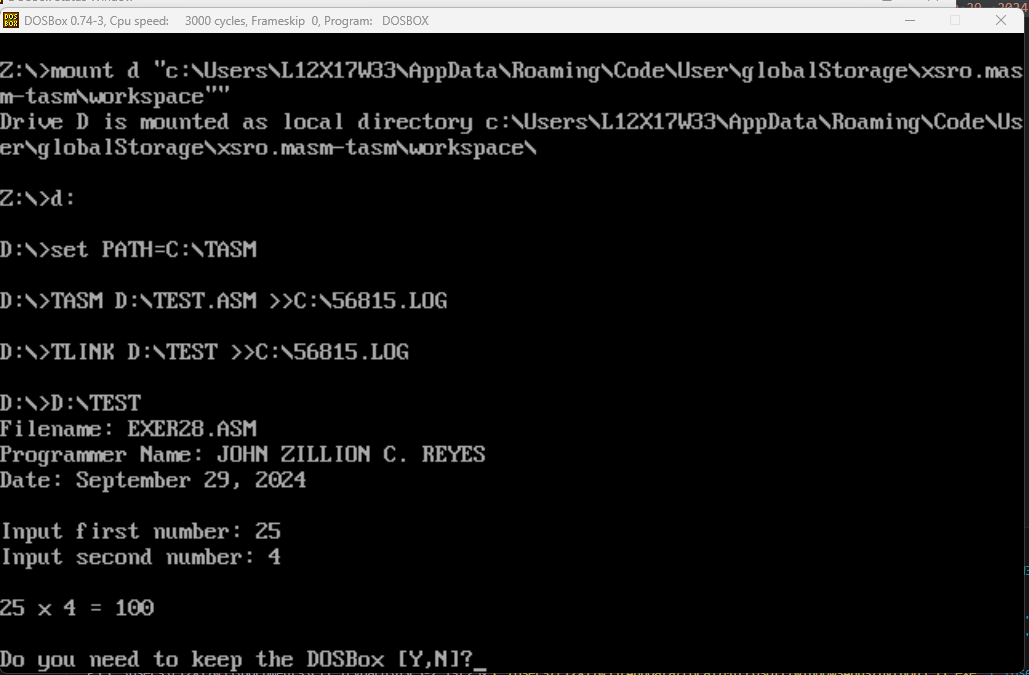
    int 21h

    pop dx

    pop ax

    ret

end start



1. EXER29.ASM

; Filename: EXER29.ASM

; Programmer Name: JOHN ZILLION C. REYES

; Date: September 29, 2024

.model small

.stack 100

.data

    szTitle db "Filename: EXER29.ASM", 0Ah, "Programmer Name: JOHN ZILLION C. REYES", 0Ah, "Date: September 29, 2024", 0Ah, 0Ah, '$'

    h1 db "Input first number: $"

    h2 db "Input second number: $"

    Op db " / $"

    Res db " = $"

    cls db "                             ", 0Dh , "                             ", 0Dh, "$"

.code

start:

    mov ax, @data

    mov ds, ax

    lea dx, szTitle

    mov ah, 09h

    int 21h

    lea di, h1

    call inputNum

    mov bx, ax

    lea di, h2

    call inputNum

    mov cx, ax

    call endLine

    mov ax, bx

    call printNum

    lea dx, Op

    call printString

    mov ax, cx

    call printNum

    xor dx, dx

    mov ax, bx

    div cx

    lea dx, Res

    call printString

    call printNum

    call endLine

    int 27h

isOdd:

    push bx

    push dx

    mov dx, 0

    mov bx, 2

    div bx

    cmp dx, 1

    pop dx

    pop bx

    ret

inputNum:

    push bx

    push cx

    push dx

    push si

    mov cx, 0

    mov si, 10

    mov bx, 0

    mov dx, di

    call printString

inputLoop:

    mov ah, 7

    int 21h

    cmp al, 8

    je inputRem

    cmp al, '0'

    jl exitInputNum

    cmp al, '9'

    jg exitInputNum

    sub al, '0'

    mov cl, al

    mov ax, bx

    mul si

    add ax, cx

    mov bx, ax

    mov dx, offset cls

    call printString

    mov dx, di

    call printString

    mov ax, bx

    call printNum

    jmp inputLoop

inputRem:

    mov ax, bx

    mov dx, 0

    div si

    mov bx, ax

    mov dx, offset cls

    call printString

    mov dx, di

    call printString

    mov ax, bx

    call printNum

    jmp inputLoop

exitInputNum:

    call endLine

    mov ax, bx

    pop si

    pop dx

    pop cx

    pop bx

    ret

printNum:

    push ax

    push bx

    push cx

    push dx

    push si

    mov cx, 0

digitLoop:

    mov bx, 10

    mov dx, 0

    div bx

    mov bx, ax

    mov ax, cx

    call isOdd

    je ifAppend

    mov ah, dl

    mov al, 0

    jmp endifAppend

ifAppend:

    pop ax

    mov al, dl

endifAppend:

    push ax

    inc cx

    mov ax, bx

    cmp ax, 0

    jne digitLoop

printLoop:

    mov ax, cx

    call isOdd

    pop ax

    je ifPrint

    mov dl, al

    push ax

    jmp endifPrint

ifPrint:

    mov dl, ah

endifPrint:

    add dl, '0'

    mov ah, 02h

    int 21h

    loop printLoop

    pop si

    pop dx

    pop cx

    pop bx

    pop ax

    ret

printString:

    push ax

    mov ah, 09h

    int 21h

    pop ax

    ret

endLine:

    push ax

    push dx

    mov ah, 02h

    mov dl, 0Ah

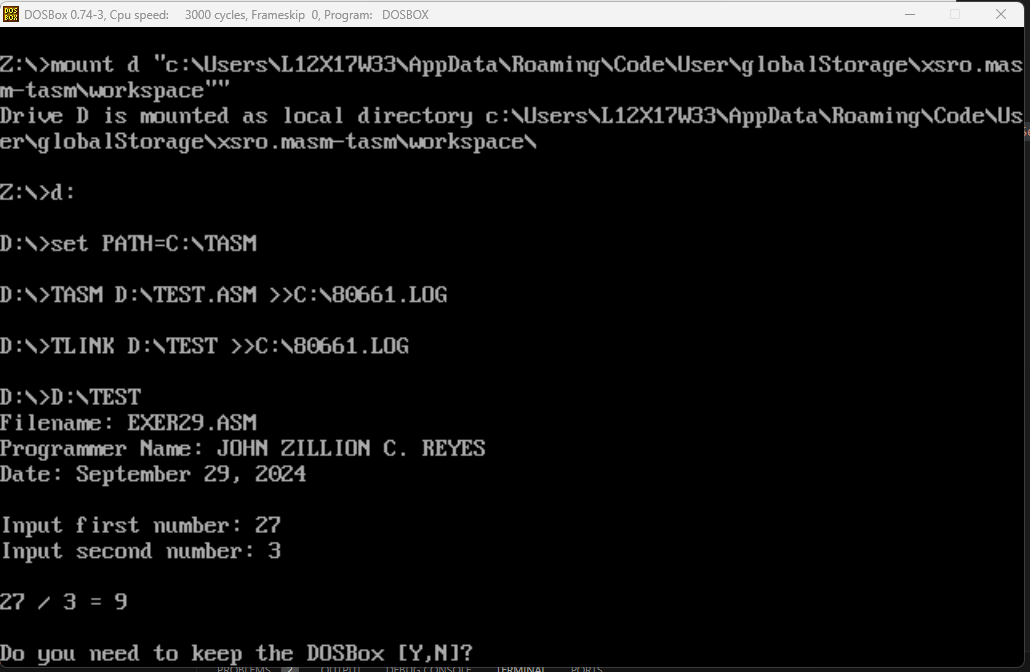
    int 21h

    pop dx

    pop ax

    ret

end start



1. EXER30.ASM

; Filename: EXER30.ASM

; Programmer Name: JOHN ZILLION C. REYES

; Date: September 29, 2024

.model small

.stack 100

.data

    szTitle db "Filename: EXER30.ASM", 0Ah, "Programmer Name: JOHN ZILLION C. REYES", 0Ah, "Date: September 29, 2024", 0Ah, 0Ah, '$'

    h1 db "Input first number: $"

    h2 db "Input second number: $"

    Great db " > $"

    Lessr db " < $"

    Equal db " = $"

    cls db "                             ", 0Dh , "                             ", 0Dh, "$"

.code

start:

    mov ax, @data

    mov ds, ax

    lea dx, szTitle

    mov ah, 09h

    int 21h

    lea di, h1

    call inputNum

    mov bx, ax

    lea di, h2

    call inputNum

    mov cx, ax

    call endLine

    mov ax, bx

    call printNum

    cmp bx, cx

    jg printGreat

    jl printLess

    lea dx, Equal

    jmp endPrint

printGreat:

    lea dx, Great

    jmp endPrint

printLess:

    lea dx, Lessr

endPrint:

    call printString

    mov ax, cx

    call printNum

    call endLine

    int 27h

isOdd:

    push bx

    push dx

    mov dx, 0

    mov bx, 2

    div bx

    cmp dx, 1

    pop dx

    pop bx

    ret

inputNum:

    push bx

    push cx

    push dx

    push si

    mov cx, 0

    mov si, 10

    mov bx, 0

    mov dx, di

    call printString

inputLoop:

    mov ah, 7

    int 21h

    cmp al, 8

    je inputRem

    cmp al, '0'

    jl exitInputNum

    cmp al, '9'

    jg exitInputNum

    sub al, '0'

    mov cl, al

    mov ax, bx

    mul si

    add ax, cx

    mov bx, ax

    mov dx, offset cls

    call printString

    mov dx, di

    call printString

    mov ax, bx

    call printNum

    jmp inputLoop

inputRem:

    mov ax, bx

    mov dx, 0

    div si

    mov bx, ax

    mov dx, offset cls

    call printString

    mov dx, di

    call printString

    mov ax, bx

    call printNum

    jmp inputLoop

exitInputNum:

    call endLine

    mov ax, bx

    pop si

    pop dx

    pop cx

    pop bx

    ret

printNum:

    push ax

    push bx

    push cx

    push dx

    push si

    mov cx, 0

digitLoop:

    mov bx, 10

    mov dx, 0

    div bx

    mov bx, ax

    mov ax, cx

    call isOdd

    je ifAppend

    mov ah, dl

    mov al, 0

    jmp endifAppend

ifAppend:

    pop ax

    mov al, dl

endifAppend:

    push ax

    inc cx

    mov ax, bx

    cmp ax, 0

    jne digitLoop

printLoop:

    mov ax, cx

    call isOdd

    pop ax

    je ifPrint

    mov dl, al

    push ax

    jmp endifPrint

ifPrint:

    mov dl, ah

endifPrint:

    add dl, '0'

    mov ah, 02h

    int 21h

    loop printLoop

    pop si

    pop dx

    pop cx

    pop bx

    pop ax

    ret

printString:

    push ax

    mov ah, 09h

    int 21h

    pop ax

    ret

endLine:

    push ax

    push dx

    mov ah, 02h

    mov dl, 0Ah

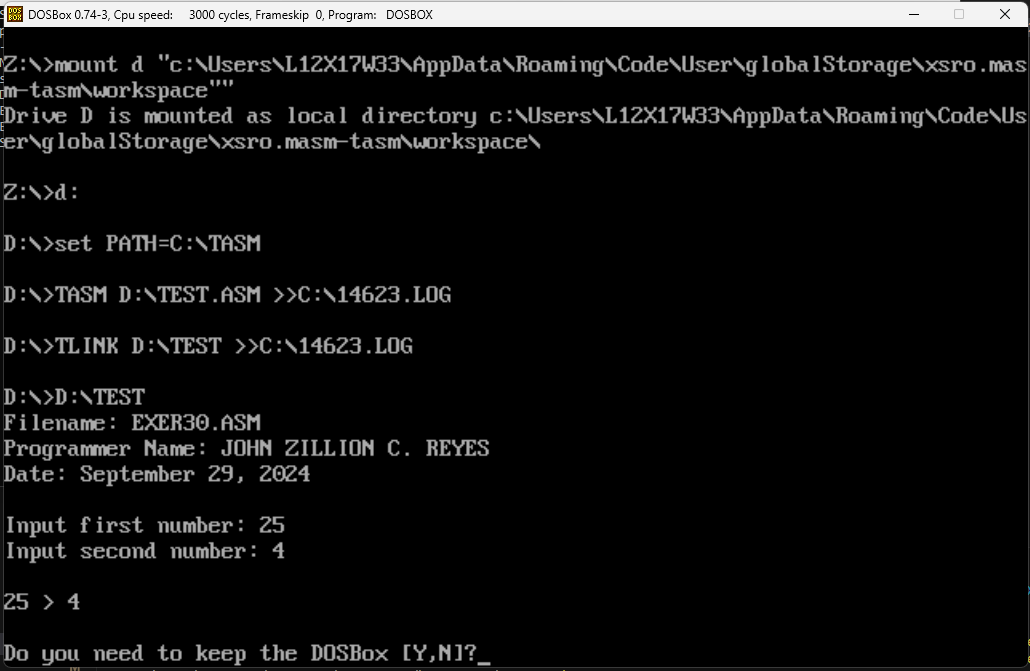
    int 21h

    pop dx

    pop ax

    ret

end start



1. EXER31.ASM

; Filename: EXER31.ASM

; Programmer Name: JOHN ZILLION C. REYES

; Date: September 29, 2024

.model small

.stack 100

.data

    szTitle db "Filename: EXER31.ASM", 0Ah, "Programmer Name: JOHN ZILLION C. REYES", 0Ah, "Date: September 29, 2024", 0Ah, 0Ah, '$'

    h1 db "Enter an integer: $"

    cls db "                             ", 0Dh , "                             ", 0Dh, "$"

.code

start:

    mov ax, @data

    mov ds, ax

    lea dx, szTitle

    mov ah, 09h

    int 21h

    lea di, h1

    call inputNum

    mov bx, ax

    mov cx, 1

numLoop:

    cmp cx, bx

    jg exitNumLoop

    mov ax, cx

    call printNum

    call endLine

    inc cx

    jmp numLoop

exitNumLoop:

    int 27h

isOdd:

    push bx

    push dx

    mov dx, 0

    mov bx, 2

    div bx

    cmp dx, 1

    pop dx

    pop bx

    ret

inputNum:

    push bx

    push cx

    push dx

    push si

    mov cx, 0

    mov si, 10

    mov bx, 0

    mov dx, di

    call printString

inputLoop:

    mov ah, 7

    int 21h

    cmp al, 8

    je inputRem

    cmp al, '0'

    jl exitInputNum

    cmp al, '9'

    jg exitInputNum

    sub al, '0'

    mov cl, al

    mov ax, bx

    mul si

    add ax, cx

    mov bx, ax

    mov dx, offset cls

    call printString

    mov dx, di

    call printString

    mov ax, bx

    call printNum

    jmp inputLoop

inputRem:

    mov ax, bx

    mov dx, 0

    div si

    mov bx, ax

    mov dx, offset cls

    call printString

    mov dx, di

    call printString

    mov ax, bx

    call printNum

    jmp inputLoop

exitInputNum:

    call endLine

    mov ax, bx

    pop si

    pop dx

    pop cx

    pop bx

    ret

printNum:

    push ax

    push bx

    push cx

    push dx

    push si

    mov cx, 0

digitLoop:

    mov bx, 10

    mov dx, 0

    div bx

    mov bx, ax

    mov ax, cx

    call isOdd

    je ifAppend

    mov ah, dl

    mov al, 0

    jmp endifAppend

ifAppend:

    pop ax

    mov al, dl

endifAppend:

    push ax

    inc cx

    mov ax, bx

    cmp ax, 0

    jne digitLoop

printLoop:

    mov ax, cx

    call isOdd

    pop ax

    je ifPrint

    mov dl, al

    push ax

    jmp endifPrint

ifPrint:

    mov dl, ah

endifPrint:

    add dl, '0'

    mov ah, 02h

    int 21h

    loop printLoop

    pop si

    pop dx

    pop cx

    pop bx

    pop ax

    ret

printString:

    push ax

    mov ah, 09h

    int 21h

    pop ax

    ret

endLine:

    push ax

    push dx

    mov ah, 02h

    mov dl, 0Ah

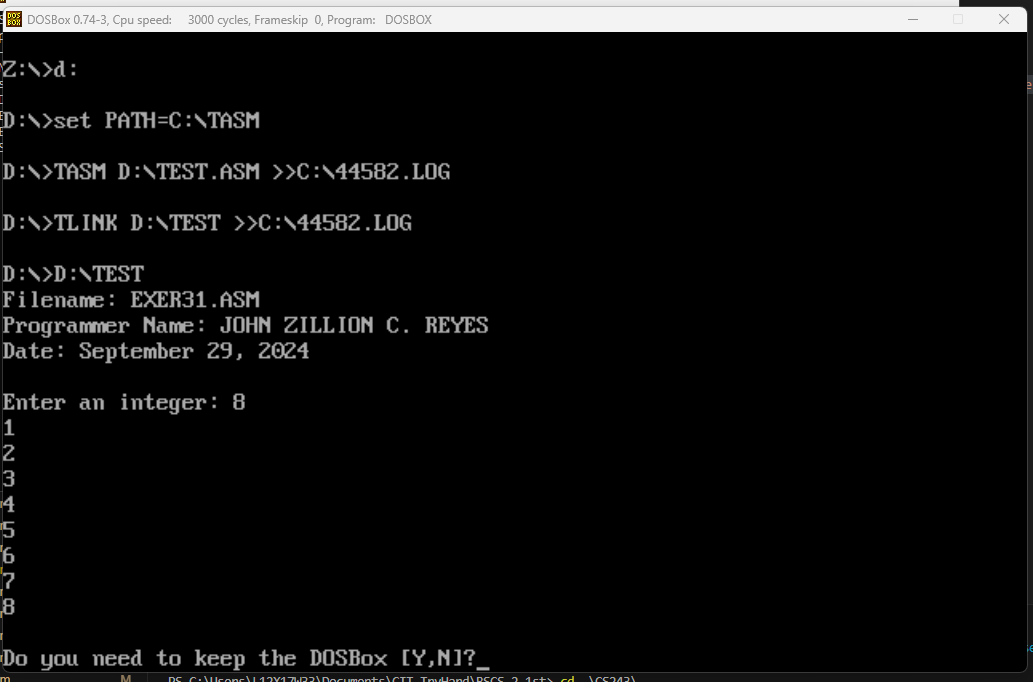
    int 21h

    pop dx

    pop ax

    ret

end start



1. EXER32.ASM

; Filename: EXER32.ASM

; Programmer Name: JOHN ZILLION C. REYES

; Date: September 29, 2024

.model small

.stack 100

.data

    szTitle db "Filename: EXER32.ASM", 0Ah, "Programmer Name: JOHN ZILLION C. REYES", 0Ah, "Date: September 29, 2024", 0Ah, 0Ah, '$'

    h1 db "THE CALCULATOR", 0Ah, "Created by: JOHN ZILLION C. REYES", 0Ah, "Date: September 28, 2024", 0Ah, 0Ah, "Main Menu", 0Ah, "$"

    hA db "a - Addition", 0Ah, "$"

    hS db "s - Subtraction", 0Ah, "$"

    hM db "m - Multiplication", 0Ah, "$"

    hD db "d - Division", 0Ah, "$"

    hE db "e - Exit", 0Ah, "Enter your choice: $"

    inA0 db "Addition", 0Ah, "$"

    inA1 db "Enter first addend: $"

    inA2 db "Enter second addend: $"

    outA1 db "First addend is: $"

    outA2 db "Second addend is: $"

    outA3 db "Sum: $"

    inS0 db "Subtraction", 0Ah, "$"

    inS1 db "Enter minuend: $"

    inS2 db "Enter subtrahend: $"

    outS1 db "Minuend is: $"

    outS2 db "Subtrahend is: $"

    outS3 db "Difference: $"

    inM0 db "Multiplication", 0Ah, "$"

    inM1 db "Enter multiplicand: $"

    inM2 db "Enter multiplier: $"

    outM1 db "Multiplicand is: $"

    outM2 db "Multiplier is: $"

    outM3 db "Product: $"

    inD0 db "Division", 0Ah, "$"

    inD1 db "Enter dividend: $"

    inD2 db "Enter divisor: $"

    outD1 db "Dividend is: $"

    outD2 db "Divisor is: $"

    outD3 db "Quotient: $"

    outE db "Exit Program", 0Ah, "$"

    outN db "INVALID CHOICE!", 0Ah, "$"

    f1 db 0Ah, "Press Enter to continue.$"

    cls db "                             ", 0Dh , "                             ", 0Dh, "$"

.code

start:

    mov ax, @data

    mov ds, ax

    lea dx, szTitle

    mov ah, 09h

    int 21h

    mov ah, 00h

    mov al, 03h

    int 10h

    lea dx, h1

    call printString

    mov bl, 60h

    mov cx, 12

    call setColor

    lea dx, hA

    call printString

    mov bl, 20h

    mov cx, 15

    call setColor

    lea dx, hS

    call printString

    mov bl, 10h

    mov cx, 18

    call setColor

    lea dx, hM

    call printString

    mov bl, 50h

    mov cx, 12

    call setColor

    lea dx, hD

    call printString

    mov bl, 70h

    mov cx, 8

    call setColor

    lea dx, hE

    call printString

    mov ah, 01h

    int 21h

    call endLine

    call endLine

    cmp al, 'a'

    jne nAdd

    call opAdd

    jmp exit

nAdd:

    cmp al, 's'

    jne nSub

    call opSub

    jmp exit

nSub:

    cmp al, 'm'

    jne nMul

    call opMul

    jmp exit

nMul:

    cmp al, 'd'

    jne nDiv

    call opDiv

    jmp exit

nDiv:

    cmp al, 'e'

    jne nEx

    mov bl, 70h

    mov cx, 12

    call setColor

    lea dx, outE

    call printString

    int 27h

nEx:

    mov cx, 15

    mov bl, 0CEh

    call setColor

    lea dx, outN

    call printString

exit:

    lea dx, f1

    call printString

    mov ah, 01h

    int 21h

    jmp start

opAdd:

    push dx

    push cx

    push bx

    push ax

    mov cx, 8

    mov bl, 60h

    call setColor

    lea dx, inA0

    call printString

    lea di, inA1

    call inputNum

    mov bx, ax

    lea di, inA2

    call inputNum

    mov cx, ax

    lea dx, outA1

    call printString

    mov ax, bx

    call printNum

    call endLine

    lea dx, outA2

    call printString

    mov ax, cx

    call printNum

    call endLine

    add ax, bx

    lea dx, outA3

    call printString

    call printNum

    call endline

    pop ax

    pop bx

    pop cx

    pop dx

    ret

opSub:

    push dx

    push cx

    push bx

    push ax

    mov cx, 11

    mov bl, 20h

    call setColor

    lea dx, inS0

    call printString

    lea di, inS1

    call inputNum

    mov bx, ax

    lea di, inS2

    call inputNum

    mov cx, ax

    lea dx, outS1

    call printString

    mov ax, bx

    call printNum

    call endLine

    lea dx, outS2

    call printString

    mov ax, cx

    call printNum

    call endLine

    sub bx, ax

    mov ax, bx

    lea dx, outS3

    call printString

    call printNum

    call endline

    pop ax

    pop bx

    pop cx

    pop dx

    ret

opMul:

    push dx

    push cx

    push bx

    push ax

    mov cx, 14

    mov bl, 10h

    call setColor

    lea dx, inM0

    call printString

    lea di, inM1

    call inputNum

    mov bx, ax

    lea di, inM2

    call inputNum

    mov cx, ax

    lea dx, outM1

    call printString

    mov ax, bx

    call printNum

    call endLine

    lea dx, outM2

    call printString

    mov ax, cx

    call printNum

    call endLine

    mul bx

    lea dx, outM3

    call printString

    call printNum

    call endline

    pop ax

    pop bx

    pop cx

    pop dx

    ret

opDiv:

    push dx

    push cx

    push bx

    push ax

    mov cx, 8

    mov bl, 50h

    call setColor

    lea dx, inD0

    call printString

    lea di, inD1

    call inputNum

    mov bx, ax

    lea di, inD2

    call inputNum

    mov cx, ax

    lea dx, outD1

    call printString

    mov ax, bx

    call printNum

    call endLine

    lea dx, outD2

    call printString

    mov ax, cx

    call printNum

    call endLine

    mov ax, bx

    mov dx, 0

    div cx

    lea dx, outD3

    call printString

    call printNum

    call endline

    pop ax

    pop bx

    pop cx

    pop dx

    ret

setColor:

    push ax

    mov ah, 09h

    int 10h

    pop ax

    ret

printString:

    push ax

    mov ah, 09h

    int 21h

    pop ax

    ret

endLine:

    push ax

    push dx

    mov ah, 02h

    mov dl, 0Ah

    int 21h

    pop dx

    pop ax

    ret

isOdd:

    push bx

    push dx

    mov dx, 0

    mov bx, 2

    div bx

    cmp dx, 1

    pop dx

    pop bx

    ret

inputNum:

    push bx

    push cx

    push dx

    push si

    mov cx, 0

    mov si, 10

    mov bx, 0

    mov dx, di

    call printString

inputLoop:

    mov ah, 7

    int 21h

    cmp al, 8

    je inputRem

    cmp al, '0'

    jl exitInputNum

    cmp al, '9'

    jg exitInputNum

    sub al, '0'

    mov cl, al

    mov ax, bx

    mul si

    add ax, cx

    mov bx, ax

    mov dx, offset cls

    call printString

    mov dx, di

    call printString

    mov ax, bx

    call printNum

    jmp inputLoop

inputRem:

    mov ax, bx

    mov dx, 0

    div si

    mov bx, ax

    mov dx, offset cls

    call printString

    mov dx, di

    call printString

    mov ax, bx

    call printNum

    jmp inputLoop

exitInputNum:

    call endLine

    mov ax, bx

    pop si

    pop dx

    pop cx

    pop bx

    ret

printNum:

    push ax

    push bx

    push cx

    push dx

    push si

    mov cx, 0

digitLoop:

    mov bx, 10

    mov dx, 0

    div bx

    mov bx, ax

    mov ax, cx

    call isOdd

    je ifAppend

    mov ah, dl

    mov al, 0

    jmp endifAppend

ifAppend:

    pop ax

    mov al, dl

endifAppend:

    push ax

    inc cx

    mov ax, bx

    cmp ax, 0

    jne digitLoop

printLoop:

    mov ax, cx

    call isOdd

    pop ax

    je ifPrint

    mov dl, al

    push ax

    jmp endifPrint

ifPrint:

    mov dl, ah

endifPrint:

    add dl, '0'

    mov ah, 02h

    int 21h

    loop printLoop

    pop si

    pop dx

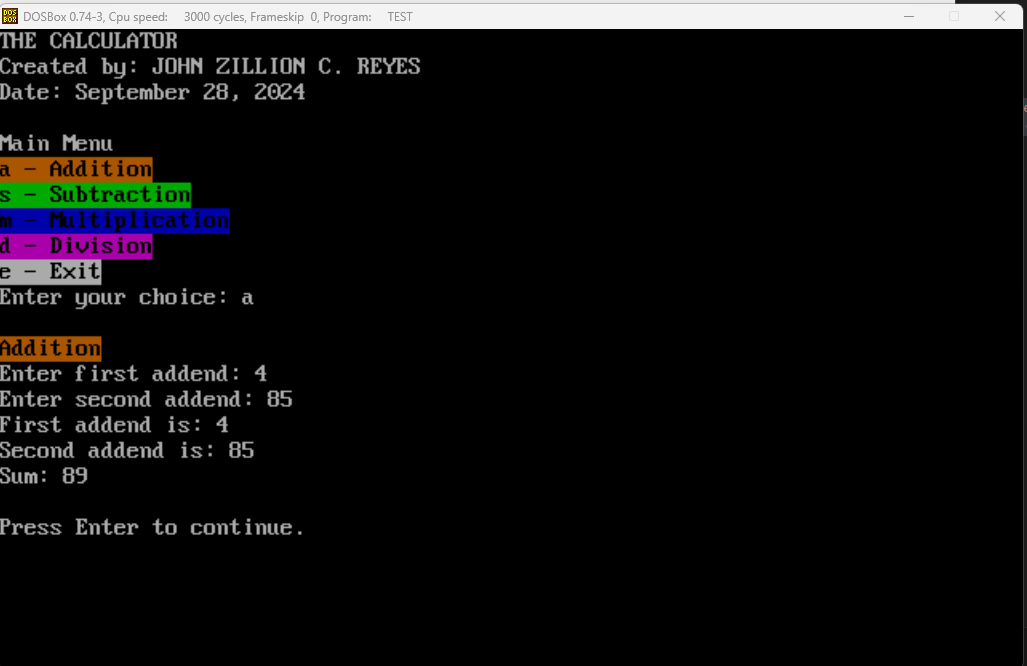
    pop cx

    pop bx

    pop ax

    ret

end start



1. Hands-on Exams
2. EXAM1.ASM

; FILENAME: EXAM1.ASM

; MULTIPLICATION TABLE

; John Zillion C. Reyes

; DATE: September 6, 2024

.model small

.stack 100

.data

    szTitle db "FILENAME: EXAM1.ASM", 0Ah, "MULTIPLICATION TABLE", 0Ah, "John Zillion C. Reyes", 0Ah, "DATE: September 6, 2024", 0Ah, 0Ah, '$'

    h1 db "MULTIPLICATION TABLE", 0Ah, "Created by: JOHN ZILLION REYES", 0Ah, "Date: September 6, 2024",  0Ah, "$"

    nx db "| X  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10  | ", 0Ah, "$"

    n1 db "| 1  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10  | ", 0Ah, "$"

    n2 db "| 2  | 2  | 4  | 6  | 8  | 10 | 12 | 14 | 16 | 18 | 20  | ", 0Ah, "$"

    n3 db "| 3  | 3  | 6  | 9  | 12 | 15 | 18 | 21 | 24 | 27 | 30  | ", 0Ah, "$"

    n4 db "| 4  | 4  | 8  | 12 | 16 | 20 | 24 | 28 | 32 | 36 | 40  | ", 0Ah, "$"

    n5 db "| 5  | 5  | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50  | ", 0Ah, "$"

    n6 db "| 6  | 6  | 12 | 18 | 24 | 30 | 36 | 42 | 48 | 54 | 60  | ", 0Ah, "$"

    n7 db "| 7  | 7  | 14 | 21 | 28 | 35 | 42 | 49 | 56 | 63 | 70  | ", 0Ah, "$"

    n8 db "| 8  | 8  | 16 | 24 | 32 | 40 | 48 | 56 | 64 | 72 | 80  | ", 0Ah, "$"

    n9 db "| 9  | 9  | 18 | 27 | 36 | 45 | 54 | 63 | 72 | 81 | 90  | ", 0Ah, "$"

    n10 db "| 10 | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 | ", 0Ah, "$"

    mulDivide db "=========================================================", 0Ah, "$"

.code

start:

    mov ax, @data

    mov ds, ax

    lea dx, szTitle

    mov ah, 09h

    int 21h

    lea dx, h1

    mov ah, 09h

    int 21h

    mov ah, 09h

    mov bl, 30h

    mov cx, 57

    int 10h

    lea dx, mulDivide

    mov ah, 09h

    int 21h

    mov ah, 09h

    mov bl, 30h

    mov cx, 57

    int 10h

    lea dx, nx

    mov ah, 09h

    int 21h

    mov ah, 09h

    mov bl, 30h

    mov cx, 5

    int 10h

    lea dx, mulDivide

    mov ah, 09h

    int 21h

    mov ah, 09h

    mov bl, 30h

    mov cx, 5

    int 10h

    lea dx, n1

    mov ah, 09h

    int 21h

    mov ah, 09h

    mov bl, 30h

    mov cx, 5

    int 10h

    lea dx, mulDivide

    mov ah, 09h

    int 21h

    mov ah, 09h

    mov bl, 30h

    mov cx, 5

    int 10h

    lea dx, n2

    mov ah, 09h

    int 21h

    mov ah, 09h

    mov bl, 30h

    mov cx, 5

    int 10h

    lea dx, mulDivide

    mov ah, 09h

    int 21h

    mov ah, 09h

    mov bl, 30h

    mov cx, 5

    int 10h

    lea dx, n3

    mov ah, 09h

    int 21h

    mov ah, 09h

    mov bl, 30h

    mov cx, 5

    int 10h

    lea dx, mulDivide

    mov ah, 09h

    int 21h

    mov ah, 09h

    mov bl, 30h

    mov cx, 5

    int 10h

    lea dx, n4

    mov ah, 09h

    int 21h

    mov ah, 09h

    mov bl, 30h

    mov cx, 5

    int 10h

    lea dx, mulDivide

    mov ah, 09h

    int 21h

    mov ah, 09h

    mov bl, 30h

    mov cx, 5

    int 10h

    lea dx, n6

    mov ah, 09h

    int 21h

    mov ah, 09h

    mov bl, 30h

    mov cx, 5

    int 10h

    lea dx, mulDivide

    mov ah, 09h

    int 21h

    mov ah, 09h

    mov bl, 30h

    mov cx, 5

    int 10h

    lea dx, n5

    mov ah, 09h

    int 21h

    mov ah, 09h

    mov bl, 30h

    mov cx, 5

    int 10h

    lea dx, mulDivide

    mov ah, 09h

    int 21h

    mov ah, 09h

    mov bl, 30h

    mov cx, 5

    int 10h

    lea dx, n7

    mov ah, 09h

    int 21h

    mov ah, 09h

    mov bl, 30h

    mov cx, 5

    int 10h

    lea dx, mulDivide

    mov ah, 09h

    int 21h

    mov ah, 09h

    mov bl, 30h

    mov cx, 5

    int 10h

    lea dx, n8

    mov ah, 09h

    int 21h

    mov ah, 09h

    mov bl, 30h

    mov cx, 5

    int 10h

    lea dx, mulDivide

    mov ah, 09h

    int 21h

    mov ah, 09h

    mov bl, 30h

    mov cx, 5

    int 10h

    lea dx, n9

    mov ah, 09h

    int 21h

    mov ah, 09h

    mov bl, 30h

    mov cx, 5

    int 10h

    lea dx, mulDivide

    mov ah, 09h

    int 21h

    mov ah, 09h

    mov bl, 30h

    mov cx, 5

    int 10h

    lea dx, n10

    mov ah, 09h

    int 21h

    mov ah, 09h

    mov bl, 30h

    mov cx, 5

    int 10h

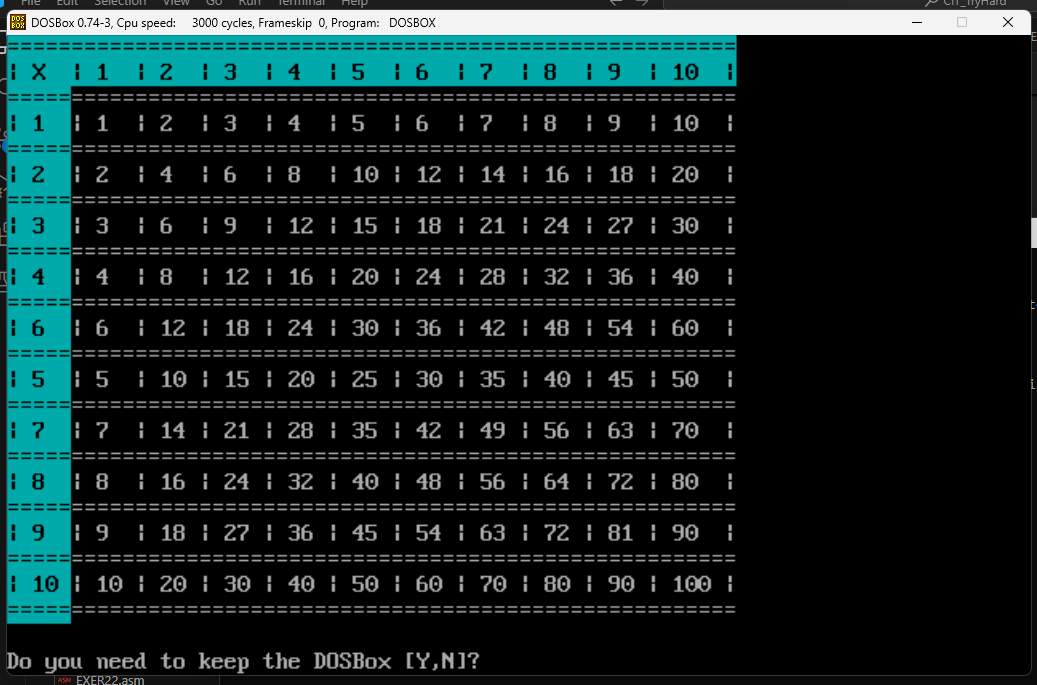
    lea dx, mulDivide

    mov ah, 09h

    int 21h

int 27h

end start



1. EXAM2.ASM

; Filename: EXAM2.ASM

; CS243 Lab Hands-on Exam No. 2

; First Semester SY 2024-2025

; Student Name: JOHN ZILLION C. REYES

; Date Finished: September 28, 2024

.model small

.stack 100

.data

    szTitle db "Filename: EXAM2.ASM", 0Ah, "CS243 Lab Hands-on Exam No. 2 ", 0Ah, "First Semester SY 2024-2025", 0Ah, "Student Name: JOHN ZILLION C. REYES", 0Ah, "Date Finished: September 28, 2024", 0Ah, 0Ah, '$'

    h1 db "THE CALCULATOR", 0Ah, "Created by: JOHN ZILLION C. REYES", 0Ah, "Date: September 28, 2024", 0Ah, 0Ah, "Main Menu", 0Ah, "$"

    hA db "a - Addition", 0Ah, "$"

    hS db "s - Subtraction", 0Ah, "$"

    hM db "m - Multiplication", 0Ah, "$"

    hD db "d - Division", 0Ah, "$"

    hE db "e - Exit", 0Ah, "Enter your choice: $"

    inA1 db "Addition", 0Ah, "Enter first addend: $"

    inA2 db "Enter second addend: $"

    outA1 db "First addend is: $"

    outA2 db "Second addend is: $"

    inS1 db "Subtraction", 0Ah, "Enter minuend: $"

    inS2 db "Enter subtrahend: $"

    outS1 db "Minuend is: $"

    outS2 db "Subtrahend is: $"

    inM1 db "Multiplication", 0Ah, "Enter multiplicand: $"

    inM2 db "Enter multiplier: $"

    outM1 db "Multiplicand is: $"

    outM2 db "Multiplier is: $"

    inD1 db "Division", 0Ah, "Enter dividend: $"

    inD2 db "Enter divisor: $"

    outD1 db "Dividend is: $"

    outD2 db "Divisor is: $"

    outE db "Exit Program", 0Ah, "$"

    outN db "INVALID CHOICE!", 0Ah, "$"

    maxLen EQU 100

    num1 db maxLen dup(?)

    num2 db maxLen dup(?)

    f1 db 0Ah, "Thank you.$"

.code

start:

    mov ax, @data

    mov ds, ax

    lea dx, szTitle

    mov ah, 09h

    int 21h

    lea dx, h1

    call printString

    mov bl, 60h

    mov cx, 12

    call setColor

    lea dx, hA

    call printString

    mov bl, 20h

    mov cx, 15

    call setColor

    lea dx, hS

    call printString

    mov bl, 10h

    mov cx, 18

    call setColor

    lea dx, hM

    call printString

    mov bl, 50h

    mov cx, 12

    call setColor

    lea dx, hD

    call printString

    mov bl, 70h

    mov cx, 8

    call setColor

    lea dx, hE

    call printString

    mov ah, 01h

    int 21h

    call endLine

    call endLine

    cmp al, 'a'

    jne nAdd

    mov cx, 8

    mov bl, 60h

    lea di, outA2

    lea dx, outA1

    lea si, inA2

    lea ax, inA1

    call printOperation

    jmp exit

nAdd:

    cmp al, 's'

    jne nSub

    mov cx, 11

    mov bl, 20h

    lea di, outS2

    lea dx, outS1

    lea si, inS2

    lea ax, inS1

    call printOperation

    jmp exit

nSub:

    cmp al, 'm'

    jne nMul

    mov cx, 14

    mov bl, 10h

    lea di, outM2

    lea dx, outM1

    lea si, inM2

    lea ax, inM1

    call printOperation

    jmp exit

nMul:

    cmp al, 'd'

    jne nDiv

    mov cx, 8

    mov bl, 50h

    lea di, outD2

    lea dx, outD1

    lea si, inD2

    lea ax, inD1

    call printOperation

    jmp exit

nDiv:

    cmp al, 'e'

    jne nEx

    mov bl, 70h

    mov cx, 12

    call setColor

    lea dx, outE

    call printString

    jmp exit

nEx:

    mov cx, 15

    mov bl, 0CEh

    call setColor

    lea dx, outN

    call printString

exit:

    lea dx, f1

    call printString

    int 27h

printOperation:

    push di

    push dx

    push si

    push ax

    call setColor

    pop bx

    lea dx, num1

    call inputNum

    pop bx

    lea dx, num2

    call inputNum

    pop bx

    lea dx, num1

    call outputNum

    pop bx

    lea dx, num2

    call outputNum

    ret

setColor:

    push ax

    mov ah, 09h

    int 10h

    pop ax

    ret

printString:

    push ax

    mov ah, 09h

    int 21h

    pop ax

    ret

endLine:

    push ax

    push dx

    mov ah, 02h

    mov dl, 0Ah

    int 21h

    pop dx

    pop ax

    ret

inputNum:

    push cx

    push dx

    mov dx, bx

    call printString

    pop dx

    mov ah, 3Fh

    mov bx, 0

    mov cx, maxLen

    int 21h

    mov si, dx

    add si, ax

    mov bl, '$'

    mov [si - 2], bl

    pop cx

    ret

outputNum:

    push dx

    mov dx, bx

    call printString

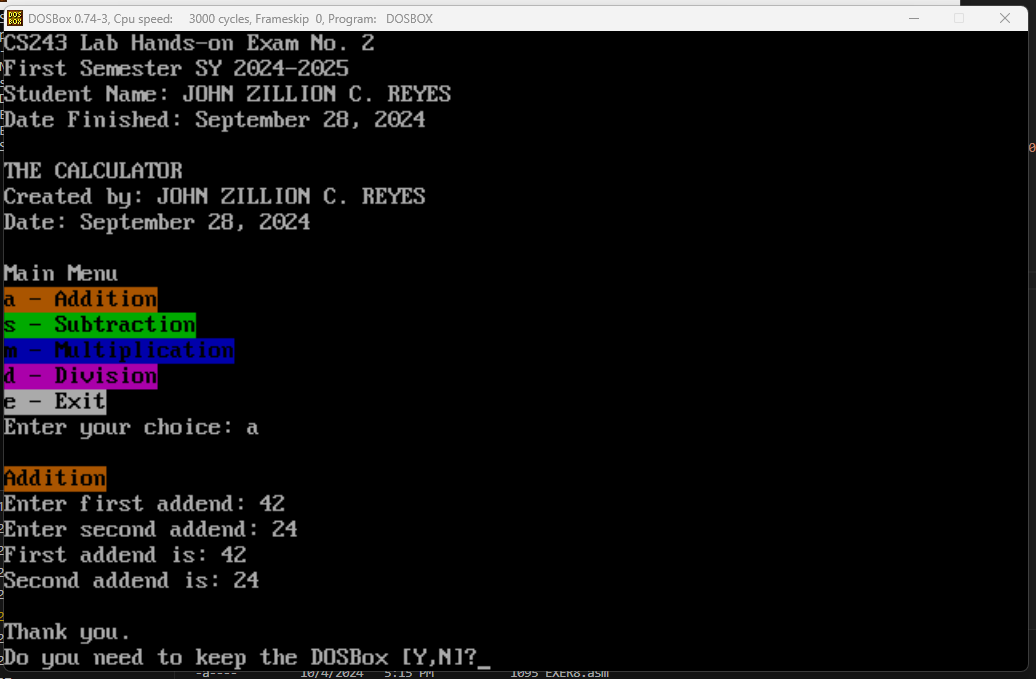
    pop dx

    call printString

    call endLine

    ret

end start



1. EXAM3.ASM

; Filename: EXAM3.ASM

; CS243 Lab Hands-on Exam No. 3

; First Semester SY 2024-2025

; Student Name: JOHN ZILLION C. REYES

; Date Finished: October 4, 2024

.model small

.stack 100

.data

    szTitle db "Filename: EXAM3.ASM", 0Ah, "CS243 Lab Hands-on Exam No. 3", 0Ah, "First Semester SY 2024-2025", 0Ah, "Student Name: JOHN ZILLION C. REYES", 0Ah, "Date Finished: October 4, 2024", 0Ah, 0Ah, '$'

    head db 1, 25, 60h, "VOTER'S REGISTRATION FORM", 0Ah, 1, 33, 20h, "Created by: JOHN ZILLION C. REYES", 0Ah, 1, 21, 50h, "Date: October 4, 2024", 0Ah, 0Ah, "Please enter the following:", 0Ah, 0Ah, 0

    i1 db "First Name: ", 0

    i2 db "Middle Name: ", 0

    i3 db "Family Name: ", 0

    i4 db "Gender: ", 0

    i5 db "Birthday", 0Ah, '  Month: ', 0

    i6 db "  Day: ", 0

    i7 db "  Year: ", 0

    i8 db "Address", 0Ah, '  House Number: ', 0

    i9 db "  Street: ", 0

    i10 db "  Barangay: ", 0

    i11 db "  City: ", 0

    i12 db "  Province: ", 0

    i13 db "Educational Background", 0Ah, "  Elementary", 0Ah, "    Name of School: ", 0

    i14 db "    Year Graduated: ", 0

    i15 db "  Junior High School", 0Ah, "    Name of School: ", 0

    i16 db "    Year Graduated: ", 0

    i17 db "  Senior High School", 0Ah, "    Name of School: ", 0

    i18 db "    Year Graduated: ", 0

    i19 db "  College", 0Ah, "    Name of School: ", 0

    i20 db "    Year Graduated: ", 0

    d1 db 200 dup(0)

    d2 db 200 dup(0)

    d3 db 200 dup(0)

    d4 db 200 dup(0)

    d5 db 200 dup(0)

    d6 db 200 dup(0)

    d7 db 200 dup(0)

    d8 db 200 dup(0)

    d9 db 200 dup(0)

    d10 db 200 dup(0)

    d11 db 200 dup(0)

    d12 db 200 dup(0)

    d13 db 200 dup(0)

    d14 db 200 dup(0)

    d15 db 200 dup(0)

    d16 db 200 dup(0)

    d17 db 200 dup(0)

    d18 db 200 dup(0)

    d19 db 200 dup(0)

    d20 db 200 dup(0)

    head2 db 0Ah, "SUMMARY OF INFORMATION", 0Ah, "Please verify if all entries are correct." , 0Ah, 0Ah, 0

    f1 db 0Ah, 1, 12, 0CEh, "Vote wisely!", 0Ah, "Thank you for registering.", 0Ah, 0

.code

printColorString:

    push ax

    push bx

    push cx

    push dx

printColorLoop:

    mov bl, 0

    cmp [si], bl

    je printColorExit

    mov bl, 1

    cmp [si], bl

    jne notColor

    mov ah, 09h

    inc si

    mov cl, [si]

    mov ch, 0

    inc si

    mov bl, [si]

    int 10h

    inc si

    jmp printColorLoop

notColor:

    mov ah, 02h

    mov dl, [si]

    int 21h

    inc si

    jmp printColorLoop

printColorExit:

    pop dx

    pop cx

    pop bx

    pop ax

    ret

inputData:

    push ax

    push bx

    push cx

    push dx

    push si

    mov si, dx

    call printColorString

    mov ah, 3FH

    mov bx, 0

    mov cx, 200

    mov dx, di

    int 21h

    mov bl, 0

    add di, ax

    mov [di - 2], bl

    pop si

    pop dx

    pop cx

    pop bx

    pop ax

    ret

outputData:

    push si

    push dx

    push ax

    mov si, dx

    call printColorString

    mov si, di

    call printColorString

    mov ah, 02h

    mov dl, 0Ah

    int 21h

    pop ax

    pop dx

    pop si

    ret

ProgramStart:

    mov ax, @data

    mov ds, ax

    lea dx, szTitle

    mov ah, 09h

    int 21h

    lea si, head

    call printColorString

    mov dx, offset i1

    mov di, offset d1

    call inputData

    mov dx, offset i2

    mov di, offset d2

    call inputData

    mov dx, offset i3

    mov di, offset d3

    call inputData

    mov dx, offset i4

    mov di, offset d4

    call inputData

    mov dx, offset i5

    mov di, offset d5

    call inputData

    mov dx, offset i6

    mov di, offset d6

    call inputData

    mov dx, offset i7

    mov di, offset d7

    call inputData

    mov dx, offset i8

    mov di, offset d8

    call inputData

    mov dx, offset i9

    mov di, offset d9

    call inputData

    mov dx, offset i10

    mov di, offset d10

    call inputData

    mov dx, offset i11

    mov di, offset d11

    call inputData

    mov dx, offset i12

    mov di, offset d12

    call inputData

    mov dx, offset i13

    mov di, offset d13

    call inputData

    mov dx, offset i14

    mov di, offset d14

    call inputData

    mov dx, offset i15

    mov di, offset d15

    call inputData

    mov dx, offset i16

    mov di, offset d16

    call inputData

    mov dx, offset i17

    mov di, offset d17

    call inputData

    mov dx, offset i18

    mov di, offset d18

    call inputData

    mov dx, offset i19

    mov di, offset d19

    call inputData

    mov dx, offset i20

    mov di, offset d20

    call inputData

    lea si, head2

    call printColorString

    mov dx, offset i1

    mov di, offset d1

    call outputData

    mov dx, offset i2

    mov di, offset d2

    call outputData

    mov dx, offset i3

    mov di, offset d3

    call outputData

    mov dx, offset i4

    mov di, offset d4

    call outputData

    mov dx, offset i5

    mov di, offset d5

    call outputData

    mov dx, offset i6

    mov di, offset d6

    call outputData

    mov dx, offset i7

    mov di, offset d7

    call outputData

    mov dx, offset i8

    mov di, offset d8

    call outputData

    mov dx, offset i9

    mov di, offset d9

    call outputData

    mov dx, offset i10

    mov di, offset d10

    call outputData

    mov dx, offset i11

    mov di, offset d11

    call outputData

    mov dx, offset i12

    mov di, offset d12

    call outputData

    mov dx, offset i13

    mov di, offset d13

    call outputData

    mov dx, offset i14

    mov di, offset d14

    call outputData

    mov dx, offset i15

    mov di, offset d15

    call outputData

    mov dx, offset i16

    mov di, offset d16

    call outputData

    mov dx, offset i17

    mov di, offset d17

    call outputData

    mov dx, offset i18

    mov di, offset d18

    call outputData

    mov dx, offset i19

    mov di, offset d19

    call outputData

    mov dx, offset i20

    mov di, offset d20

    call outputData

    lea si, f1

    call printColorString

    int 27h

end ProgramStart

