**מעבדה 7**

**מגישים:**

טל יהושע, ת"ז 314774860

משה פרץ, ת"ז 318263977

**תרגיל מס' 1:**

**תוכנית:**

;

; This program gets from the user two 4 bytes number and checks if the Amicable

;

.MODEL SMALL

.STACK 100h

.DATA

Num DB 12 DUP ('$')

TEN DD 10

Menu DB 13,10,'Please enter 2 numbers:',13,10,'$'

Amicable DB 13,10,'Amicable Numbers',13,10,'$'

NotAmicable DB 13,10,'Not Amicable Numbers',13,10,'$'

.CODE

.386

; Function name: \_getNum

; this function get input from the user (number) and return it by EAX

;

;

\_getNum PROC NEAR ; output AX

PUSH BP

MOV BP, SP ; set BP so it can use the inputs

PUSH CX

MOV DX, OFFSET Num

MOV CX, 12

MOV AH, 3Fh

MOV BX, 0

INT 21h ; get the number

LEA DI, Num

MOV CX, AX

SUB CX, 2 ; get the real length on the number (minus the enter)

XOR EAX, EAX

XOR EBX, EBX

lop: ; get the number from the string

MOV BL, [DI]

SUB BL, '0'

MUL TEN

ADD EAX, EBX

INC DI

LOOP lop

POP CX

POP BP

RET

\_getNum ENDP

; Function name: \_PNum

; This function get number and return the sum of his dividers

;

;

\_PNum PROC NEAR ; input: number

PUSH ECX

PUSH BP

MOV BP, SP

MOV ECX, [BP+8] ; get the number from stack

XOR EBX, EBX

SHR ECX, 1 ; divide it by 2 (max divider can be num/2)

CMP ECX, 0 ; if num/2 = 0 the sum is zero

JE AfterLoop

LOOOP:

MOV EAX, [BP+8] ; get the number for checking if ECX % EAX(num) = 0

XOR EDX, EDX

DIV ECX

CMP EDX, 0

JNE Skip ; if we dont have remainder

ADD EBX, ECX ; we add the the sum (EBX) the value of the divisor (ECX)

Skip:

LOOP LOOOP

AfterLoop:

MOV EAX, EBX ; return the sum

POP BP

POP ECX

RET

\_PNum ENDP

Dos\_Print PROC NEAR

MOV AH, 9h

INT 21h

RET

Dos\_Print ENDP

Main:

MOV AX, @DATA

MOV DS, AX

MOV DX, OFFSET Menu ; output Menu string

CALL Dos\_Print

CALL \_getNum

PUSH EAX ; push the value of the first number to stack

CALL \_PNum

PUSH EAX ; push the sum of the dividers for the first num

CALL \_getNum

PUSH EAX ; push the value of the second number to stack

MOV BP, SP

CMP EAX, [BP + 4] ; compare the value of the sum for the first num with the value of the second num

JNE NotAm ; if they dont match, its not amicable numbers, jump to NotAm label

CALL \_PNum ;in EAX the sum of the diriders for the second num

CMP EAX, [BP + 8] ; compare the value of the sum for the second num with the value of the first num

JNE NotAm

; if we here, they are amicable numbers

MOV DX, OFFSET Amicable

CALL Dos\_Print

JMP EndPro

NotAm: ; if we here, they are not amicable numbers

MOV DX, OFFSET NotAmicable

CALL Dos\_Print

EndPro:

ADD SP, 12 ; release stack

MOV AH,4Ch ; Set terminate option for int 21h

    INT 21h ; Return to DOS (terminate program)

END Main

**פלטים:**





