; macro for a simple addition operation

; system calls defined

SYS\_IN equ 0

SYS\_EXIT equ 1

SYS\_OUT equ 1

SYS\_READ equ 3

SYS\_WRITE equ 4

; macro for displaying output

%macro output 2 ; takes in two parameter

mov eax,SYS\_WRITE

mov ebx, SYS\_OUT

mov ecx, %1 ; parameter 1

mov edx,%2 ; parameter 2

int 80h

%endmacro

; macro for taking in output

%macro input 2

mov eax,SYS\_READ

mov ebx, SYS\_IN

mov ecx, %1

mov edx,%2

int 80h

%endmacro

; macro for adding the numbers

%macro adder 2

mov eax, [%1] ; load in first number

sub eax, '0' ; remove zero for proper format manipulation

mov ebx, [%2] ; load in second number

sub eax, '0' ; remove zero for proper format manipulation

add eax, ebx ; perform addition

add eax, '0' ; add back zero to dispaly in binary

mov [result], eax ;tranfer accumulator contents into variable result

%endmacro

section .bss

;reserve memory space for numbers and results

firstnum resb 1

secondnum resb 2

result resb 2

section .text

global \_start

\_start:

output message, messageLength ; call output to display message

input firstnum, 2 ; call input to receive first number

output messageTwo, messageTwoLength ; ask for second number

input secondnum, 2 ; receive second number

output messageThree, messageThreeLength ; message to show result is coming

adder firstnum, secondnum ; call adder to perform addition of the two numbers

int 80h

output result, 2 ; display the result

int 80h

mov eax, SYS\_EXIT

int 80h

section .data

; all messages and lengths defined here

message db 'Enter the first number',0xa

messageLength equ $ - message

messageTwo db 'Enter the second number',0xa

messageTwoLength equ $ - messageTwo

messageThree db 'The result is', 0xa

messageThreeLength equ $ - messageThree