;nasm 2.13.02

section .data

nLine: db 0xA, 0xD

nLineLen: equ $-nLine

spce: db ' ',10

plos: db '+',10

minu: db '-',10

mult: db '\*',10

abc: db '/',10

bca: db '=',10

lolbert: db 'Invalid input (negative subtraction)',10

lolLen: equ $-lolbert

section .bss

var: resb 1

var1: resb 1

var2: resb 1

reslt: resb 1

section .text

global \_start

\_start:

read:

;reading into var

mov eax,3

mov ebx,0

mov ecx,var

mov edx,1

int 80h

mov eax,3

mov ebx,0

mov ecx,var1

mov edx,1

int 80h

mov eax,3

mov ebx,0

mov ecx,var2

mov edx,1

int 80h

cmp [var],byte 46 ;comparing var to '.'

je end ;if var is '.', end program

call Sign

mov eax,3

mov ebx,0

mov ecx,var

mov edx,2

int 80h

mov eax,4

mov ebx,1

mov ecx,nLine

mov edx,nLineLen

int 80h

jmp read

Sign:

cmp [var1],byte '+'

je dda

cmp [var1],byte '-'

je minus

cmp [var1],byte '\*'

je multiply

cmp [var1],byte '/'

je divide

jmp end

ret

dda:

mov eax,4

mov ebx,1

mov ecx,var

mov edx,1

int 80h

mov eax,4

mov ebx,1

mov ecx,plos

mov edx,1

int 80h

mov eax,4

mov ebx,1

mov ecx,var2

mov edx,1

int 80h

mov eax,4

mov ebx,1

mov ecx,bca

mov edx,1

int 80h

mov al,[var2]

mov bl,[var]

add al,bl

mov [reslt],al

sub [reslt],byte '0'

int 80h

mov eax,4

mov ebx,1

mov ecx,reslt

mov edx,1

int 80h

ret

minus:

mov al,[var]

sub al,'0'

mov bl,[var2]

sub bl,'0'

sub al,bl

mov [reslt],al

add [reslt],byte '0'

cmp [reslt],byte '0'

jl invlid

mov eax,4

mov ebx,1

mov ecx,var

mov edx,1

int 80h

mov eax,4

mov ebx,1

mov ecx,minu

mov edx,1

int 80h

mov eax,4

mov ebx,1

mov ecx,var2

mov edx,1

int 80h

mov eax,4

mov ebx,1

mov ecx,bca

mov edx,1

int 80h

mov eax,4

mov ebx,1

mov ecx,reslt

mov edx,1

int 80h

ret

divide:

mov bl,[var2]

sub bl,'0'

mov al,[var]

sub al,'0'

div bl

mov [reslt],al

add [reslt],byte '0'

int 80h

mov eax,4

mov ebx,1

mov ecx,var

mov edx,1

int 80h

mov eax,4

mov ebx,1

mov ecx,abc

mov edx,1

int 80h

mov eax,4

mov ebx,1

mov ecx,var2

mov edx,1

int 80h

mov eax,4

mov ebx,1

mov ecx,bca

mov edx,1

int 80h

mov eax,4

mov ebx,1

mov ecx,reslt

mov edx,1

int 80h

jmp end

ret

multiply:

mov eax,4

mov ebx,1

mov ecx,var

mov edx,1

int 80h

mov eax,4

mov ebx,1

mov ecx,mult

mov edx,1

int 80h

mov eax,4

mov ebx,1

mov ecx,var2

mov edx,1

int 80h

mov eax,4

mov ebx,1

mov ecx,bca

mov edx,1

int 80h

mov al,[var2]

sub al,'0'

mov bl,[var]

sub bl,'0'

mul bl

mov [reslt],al

add [reslt],byte '0'

int 80h

mov eax,4

mov ebx,1

mov ecx,reslt

mov edx,1

int 80h

ret

invlid:

mov eax,4

mov ebx,1

mov ecx,lolbert

mov edx,lolLen

int 80h

end:

mov eax,1

mov ebx,0

int 80h