8.

bits 32 ; assembling for the 32 bits architecture

; (a+b-d)+(a-b-d) a - byte, b - word, c - double word, d - qword

; declare the EntryPoint (a label defining the very first instruction of the program)

global start

; declare external functions needed by our program

extern exit ; tell nasm that exit exists even if we won't be defining it

import exit msvcrt.dll ; exit is a function that ends the calling process. It is defined in msvcrt.dll

; msvcrt.dll contains exit, printf and all the other important C-runtime specific functions

; our data is declared here (the variables needed by our program)

segment data use32 class=data

; ...

a db 8

b dw 13

c dd 25

d dq 11

; our code starts here

segment code use32 class=code

start:

; ...

mov AL, [a]; al=8

cbw ;ax=8

add AX, [b];ax=a+b

cwde

cdq edx:eax=a+b

mov EBX, dword[d]

mov ECX, dword[d+4] ; ecx:ebx=d

sub EAX, EBX

sbb EDX, ECX ; edx:eax= a+b-d

mov ECX, EDX

mov EBX, EAX; ecx:ebx=a+b-d

mov AL, [a]

cbw ; ax=a

sub AX, [b]; ax=a-b

cwde

cdq ; edx:eax=a-b

sbb EDX, dword[d+4]

sub EAX, dword[d]; edx:eax=a-b-d

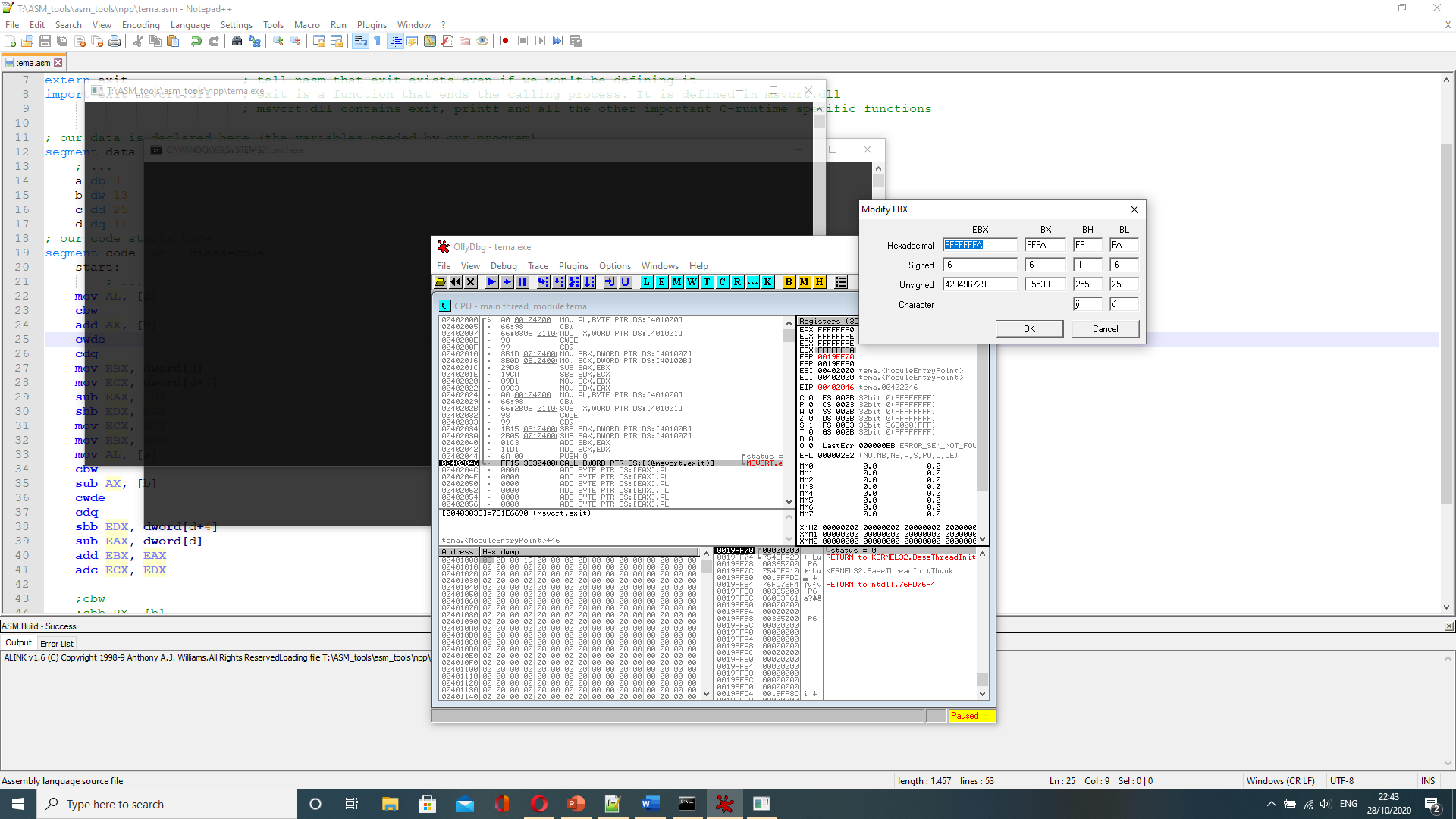
add EBX, EAX

adc ECX, EDX; ecx:ebx= (a+b-d)+(a-b-d)

; exit(0)

push dword 0 ; push the parameter for exit onto the stack

call [exit] ; call exi t to terminate the program



8.

bits 32 ; assembling for the 32 bits architecture

;(b+c+d)-(a+a) a - byte, b - word, c - double word, d - qword

; declare the EntryPoint (a label defining the very first instruction of the program)

global start

; declare external functions needed by our program

extern exit ; tell nasm that exit exists even if we won't be defining it

import exit msvcrt.dll ; exit is a function that ends the calling process. It is defined in msvcrt.dll

; msvcrt.dll contains exit, printf and all the other important C-runtime specific functions

; our data is declared here (the variables needed by our program)

segment data use32 class=data

; ...

a db 8

b dw 13

c dd 25

d dq 11

; our code starts here

segment code use32 class=code

start:

; ...

mov AX, [b]

cwde ;eax=b

add EAX, [c];eax=b+c

cdq;edx;eax=b+c

add EAX, dword[d]

adc EDX, dword[d+4]; edx:eax=b+c+d

mov ECX, EAX

mov EBX, EDX; ebx:ecx=b+c+d

mov AL, [a]; al=a

add AL, [a]; al=a+a

cbw; ax=a+a

cwde

cdq; edx:eax=a+a

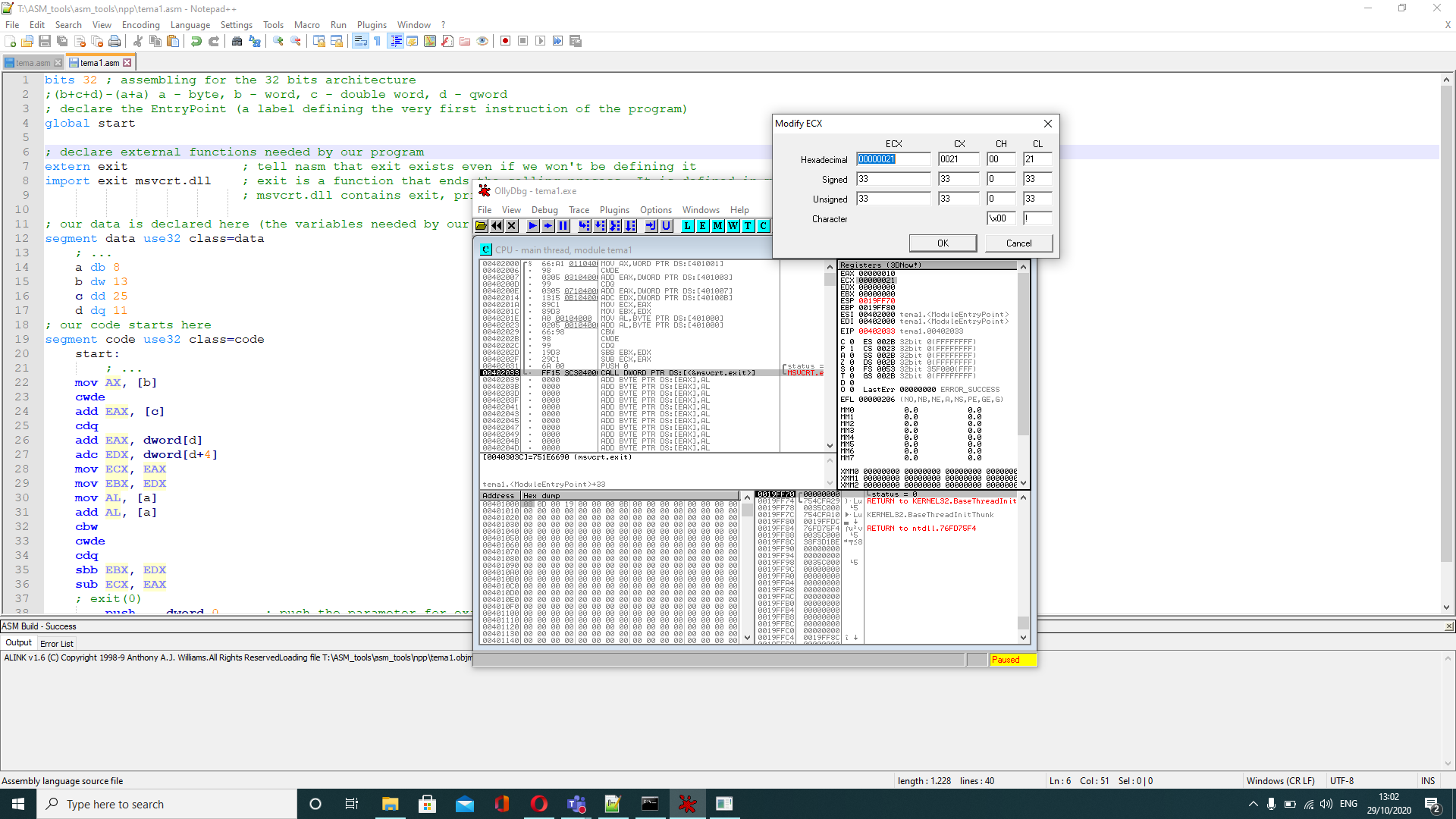
sbb EBX, EDX

sub ECX, EAX; ebx:ecx=(b+c+d)-(a+a)

; exit(0)

push dword 0 ; push the parameter for exit onto the stack

call [exit] ; call exit to terminate the program



8.

bits 32 ; assembling for the 32 bits architecture

;1/a+200\*b-c/(d+1)+x/a-e; a,b-word; c,d-byte; e-doubleword; x-qword

; declare the EntryPoint (a label defining the very first instruction of the program)

global start

; declare external functions needed by our program

extern exit ; tell nasm that exit exists even if we won't be defining it

import exit msvcrt.dll ; exit is a function that ends the calling process. It is defined in msvcrt.dll

; msvcrt.dll contains exit, printf and all the other important C-runtime specific functions

; our data is declared here (the variables needed by our program)

segment data use32 class=data

; ...

a dw 2

b dw 1

c db 4

d db 7

e dd 121

x dq 2

y resd 1

; our code starts here

segment code use32 class=code

start:

; ...

mov AL, 1

cbw

cwd

idiv word[a]; ax=1/a

cwde;eax=1/a

mov EBX, EAX

mov AL, 200

cbw

imul word[b];eax=200\*b

adc EAX, EBX

mov EBX, EAX;ebx=1/a+200\*b

mov CL, [c]

mov CH, [d]

add CH, 1

idiv CH; cx=c/(d+1)

cwd;ecx=c/(d+1)

sub EBX, ECX; ebx=1/a+200\*b-c/(d+1)

;cwde ; ebx are primele 2 operatii

mov EDX, dword[x]

mov EAX, dword[x+4]; edx;eax=x

mov CX, [a]

cwde

idiv ECX; eax=x/a

adc EBX, EAX; eax=1/a+200\*b-c/(d+1)+x/a

mov ECX, [e]

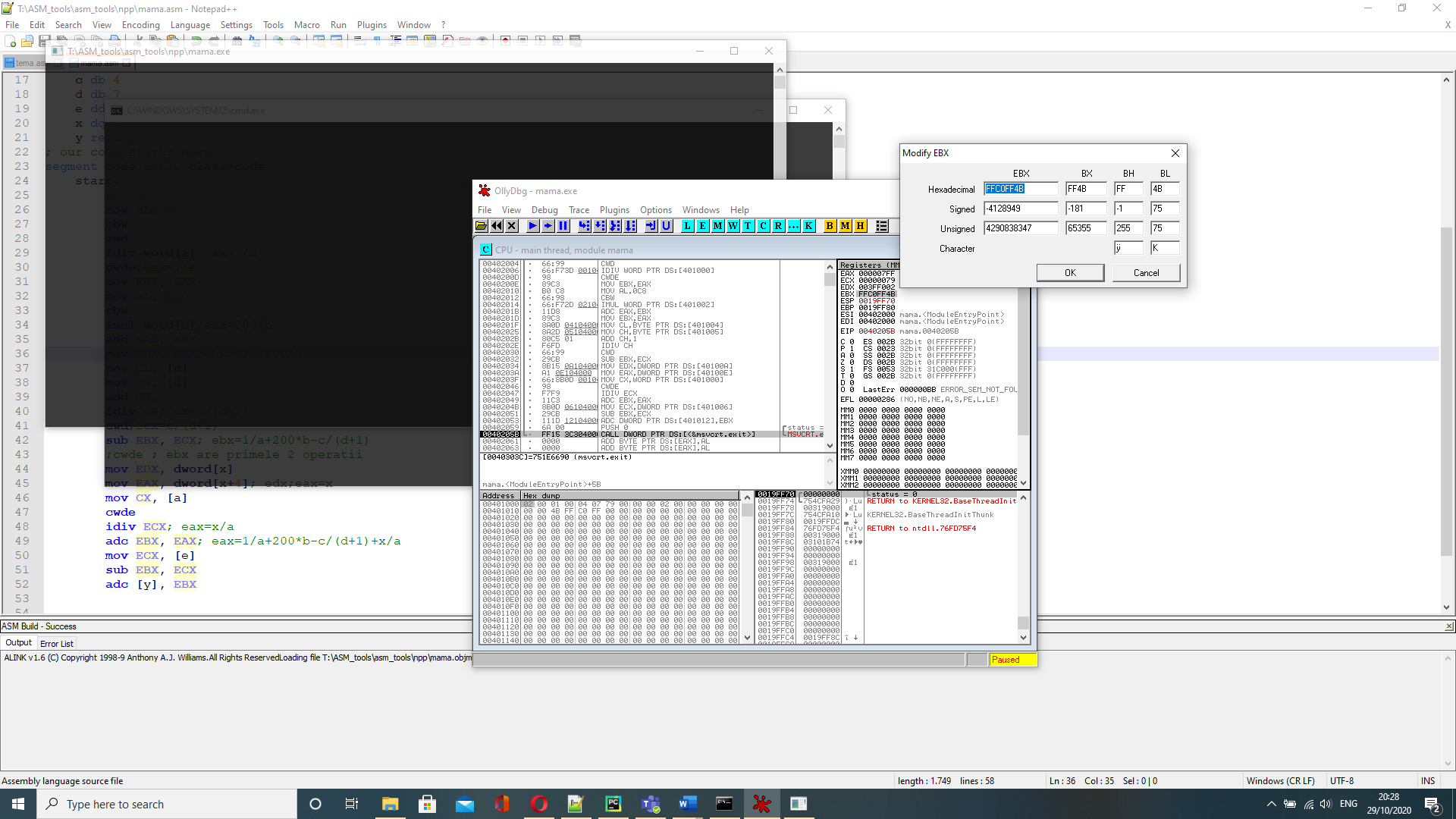
sub EBX, ECX

adc [y], EBX

; exit(0)

push dword 0 ; push the parameter for exit onto the stack

call [exit] ; call exit to terminate the program



8. unsigned

bits 32 ; assembling for the 32 bits architecture

; (a+b-d)+(a-b-d) a - byte, b - word, c - double word, d - qword

; declare the EntryPoint (a label defining the very first instruction of the program)

global start

; declare external functions needed by our program

extern exit ; tell nasm that exit exists even if we won't be defining it

import exit msvcrt.dll ; exit is a function that ends the calling process. It is defined in msvcrt.dll

; msvcrt.dll contains exit, printf and all the other important C-runtime specific functions

; our data is declared here (the variables needed by our program)

segment data use32 class=data

; ...

a db 8

b dw 13

c dd 25

d dq 5

; our code starts here

segment code use32 class=code

start:

; ...

mov AX, [a]; ax=a

add AX, [b];ax=a+b

sub AX, [d];ax=a+b-d

add AX, [a];ax=a+b-d+a

sub AX, [b]:ax=a+b-d+a-b

sub AX, [d];ax=(a+b-d)+(a-b-d)

; exit(0)

push dword 0 ; push the parameter for exit onto the stack

call [exit] ; call exi t to terminate the program

