Adunari,scaderi fara semn Pb18 - (d+d)-a-b-c

bits 32 ; assembling for the 32 bits architecture

; 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 5

b dw 8192

c dd 4194304

d dq 137438953472

;(d+d)-a-b-c=(137438953472+137438953472)-5-8192-4194304=274877906944-5-8192-4194304=274873704443=3FFFBFDFFB

; our code starts here

segment code use32 class=code

start:

;EDX:EAX=d

mov EAX,dword[d] ;EAX=0h

mov EDX,dword[d+4] ;EDX:EAX=137438953472=00000020 00000000h

mov EBX,dword[d] ;EBX=0h

mov ECX,dword[d+4] ;ECX:EBX=137438953472=00000020 00000000h

add EAX,EBX ;EAX=EAX+EBX=0h

adc EDX,ECX ;EDX=EDX+ECX+CF=40h

;EDX:EAX=d+d=00000040 00000000h

;convertim byte[a] la quadword

mov EBX,0

mov ECX,0

mov BL,[a] ;ECX:EBX=00000000 00000005h

sub EAX,EBX ;EAX=EAX-EBX

sbb EDX,ECX ;EDX=EDX-ECX-CF, EDX:EAX=(d+d)-a=274877906939=0000003F FFFFFFFBh

;convertim word[b] la quadword

mov EBX,0

mov ECX,0

mov BX,[b] ;ECX:EBX=8192=00000000 00002000h

sub EAX,EBX ;EAX=EAX-EBX

sbb EDX,ECX ;EDX=EDX-ECX-CF, EDX:EAX=(d+d)-a-b=274877898747=0000003F FFFFDFFBh

;convertim dword[c] la quadword

mov EBX,[c]

mov ECX,0 ;ECX:EBX=000000 00400000h

sub EAX,EBX ;EAX=EAX-EBX

sbb EDX,ECX ;EDX=EDX-ECX-CF, EDX:EAX=(d+d)-a-b-c=274869510139=0000003F FFBFDFFBh

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

call [exit] ; call exit to terminate the program

Graphical user interface, text, application

Description automatically generatedDiagram

Description automatically generated

Adunari si scaderi cu semn Pb18 - (d-b)-a-(b-c)

bits 32 ; assembling for the 32 bits architecture

; 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

; msvcrt.dll contains exit, printf and all the other important C

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

segment data use32 class=data

a db -5

b dw 8192

c dd 4194304

d dq 137438953472

;(d-b)-a-(b-c)=(137438953472-8192)-(-5)-(8192-4194304)=137443131397=00000020 003FC005h

; our code starts here

segment code use32 class=code

start:

;convertim word-ul b la quadword EDX:EAX

mov AX,[b] ;AX=b

cwde

cdq ;EDX:EAX=b=8192=00000000 00002000h

;punem quadword-ul d in ECX:EBX

mov EBX,dword[d]

mov ECX,dword[d+4] ;ECX:EBX=d=137438953472=000000020 00000000h

sub EBX,EAX ;EBX=EBX-EAX

sbb ECX,EDX ;ECX=ECX-EDX-CF=d-b=137438945280=0000001F FFFFE000h

;convertim byte-ul a la quadword EDX:EAX

mov AL,[a]

cbw

cwde

cdq ;EDX:EAX=-5=FFFFFFFF FFFFFFFBh

sub EBX,EAX ;EBX=EBX-EAX

sbb ECX,EDX ;ECX=ECX-EDX-CF=(d-b)-a=137438945285=0000001F FFFFE005h

;convertim word-ul b la doubleword EAX

mov AX,[b]

cwde

mov EDX,[c] ;punem in EDX valoarea lui c

sub EAX,EDX ;EAX=EAX-EDX=(b-c)=-4186112=FFFFFFFF FFC02000h

;convertim dw din EAX la quadword EDX:EAX

cdq ;EAX->EDX:EAX

sub EBX,EAX ;EBX=EBX-EAX

sbb ECX,EDX ;ECX=ECX-EDX-CF=(d-b)-a-(b-c)=137443131397=00000020 003FC005h

;rezultatul este in ECX:EBX

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

call [exit] ; call exit to terminate the program

Graphical user interface, text, application

Description automatically generatedDiagram

Description automatically generated with medium confidence

Inmultiri si impartiri Pb.18 - (a+b\*c+2/c)/(2+a)+e+x

bits 32 ; assembling for the 32 bits architecture

; 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 contains exit, printf and all the other important C

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

segment data use32 class=data

a db 5

b db -10

c dw 8192

e dd 4194304

x dq 137438953472

;(a+b\*c+2/c)/(2+a)+e+x=(5+(-10)\*8192+2/8192)/(2+5)+4194304+137438953472=137443136074=00000020 003FD24Ah

; our code starts here

segment code use32 class=code

start:

;b\*c

mov AL,[b] ;AL=b=-10

cbw ;AX=-10

imul word[c] ;DX:AX=AX\*c=-10\*8192=-81920=FFFEC000h

;mutam rezultatul in registrul EBX

push DX

push AX

pop EBX ;EBX=-81920

mov AL,[a] ;AL=5

cbw ;AX=5

cwde ;EAX=4

;a+b\*c

add EBX,EAX ;EBX=EBX+EAX=a+b\*c=-81915=FFFEC005h

mov AL,2 ;AL=2

cbw ;AX=2

cwd ;DX:AX=2

idiv word[c] ;AX=DX:AX/c=2/c=0

cwde ;EAX=AX=0

add EBX,EAX ;EBX=EBX+EAX=a+b\*c+2/c=-81915=FFFEC005h

mov AL,[a] ;AL=a=5

add AL,2 ;AL=AL+2=a+2=2+a=7

cbw ;AX=AL=7

mov CX,AX ;CX=AX=7

mov EAX,EBX ;EAX=EBX=-81915

;se "sparge" EAX-ul in 2 DX:AX pentru impartire

push EAX

pop AX

pop DX

idiv CX ;AX=DX:AX/CX=(a+b\*c+2/c)/(2+a)=-81915/7=-11702=FFFFD24Ah

cwde ;EAX=AX

add EAX,dword[e] ;EAX=EAX+e=(a+b\*c+2/c)/(2+a)+e=-11702+4194304=4182602=003FD24Ah

cdq

add EAX,dword[x]

add EDX,dword[x+4] ;EDX:EAX=EDX:EAX+x=(a+b\*c+2/c)/(2+a)+e+x=137443136074=00000020 003FD24Ah

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

call [exit] ; call exit to terminate the program

Graphical user interface, text, application, chat or text message

Description automatically generatedText

Description automatically generated

;se "sparge" EAX-ul in 2 DX:AX pentru impartire

push EAX

pop AX

pop DX

idiv CX ;AX=DX:AX/CX=(a+b\*c+2/c)/(2+a)=-81915/7=-11702=FFFFD24Ah

cwde ;EAX=AX

add EAX,dword[e] ;EAX=EAX+e=(a+b\*c+2/c)/(2+a)+e=-11702+4194304=4182602=003FD24Ah

cdq

add EAX,dword[x]

add EDX,dword[x+4] ;EDX:EAX=EDX:EAX+x=(a+b\*c+2/c)/(2+a)+e+x=137443136074=00000020 003FD24Ah

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

call [exit] ; call exit to terminate the program

Table

Description automatically generated with low confidence

II fara semn PB7 -(a-2)/(b+c)+a\*c+e-x

bits 32 ; assembling for the 32 bits architecture

; 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 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 22

b db 6

c dw 100

e dd 10

x dq 1000

;(a-2)/(b+c)+a\*c+e-x=20/106+2200+10-1000=0+2200+10-1000=1210

; our code starts here

segment code use32 class=code

start:

;convertim a la dw in perecea DX:AX pentru impartire

mov AX,0

add AL,[a]

sub AX,2;AX=(a-2)

mov DX,0

mov BX,0

add BL,[b]

add BX,[c];BX=(b+c)

div BX ;AX=DX:AX/BX=(a-2)/(b+c)=0

mov DX,AX ;DX=AX=0

mov EBX,0

add BX,DX ;BX=DX=AX=0

mov AL,[a];AL=a=22=16h

mov AH,0 ;AX=a=22=16h

mul word[c];DX:AX=a\*c=2200

push DX

push AX

pop EAX ;EAX=DX:AX=a\*c=2200

add EAX,EBX;EAX=(a-2)/(b+c)+a\*c=2200

add EAX,[e];EAX=(a-2)/(b+c)+a\*c+e=2210

mov EDX,0

sub EAX,dword[x] ;EAX=EAX-[x]

sbb EDX,dword[x+4] ;EDX=EAX-[x]-CF=(a-2)/(b+c)+a\*c+e-x=1210=4BAh

;rezultat in EDX:EAX

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

call [exit] ; call exit to terminate the program

Diagram

Description automatically generatedGraphical user interface, text, application

Description automatically generated