POSTESCU RĂZVAN-IOAN Arg. 2. GRUPA-21

a=10
b=10
c=5
d=130

(130 - 110)/(20-3) + 100 - 5 =

= 20/17 + 95. = 95 + ~~[illegible]~~ 1?

95 + 1 = 96

```
bits 32 ; EAX = (d-110)(a+b-3) + a-b-c
global start
extern exit
import exit msvcrt.dll
segment data use 32 class = data
        a db
        b dw
        c dd
        d dd
segment code use32 class = code
start:
 mov AX, [d] ; punem partea low a lui [d] pe AX
 mov DX, [d+2]; punem partea high a lui [d] pe DX
 sub AX, 110 ; facem operatia d-110
 sbb DX, 0 ; tratăm cazul cu borrow
 push DX;
 push AX ; punem DX:AX = d-110 pe stivă pt. că
 vom folosi conversii după
 mov BX, [b] ; BX = [b]
 mov AL, [a]; AL=[a]
 cbw ; AL -> AX = [a]
 add BX, AX; facem operatia a+b
 sub BX, 3 ; facem operația a+b-3
 pop AX;
 pop DX; scoatem de pe stiva d-110
 idiv BX; efectuăm împărțirea (d-110)/(a+b-3) unde
 rezultatul se află în AX și restul în DX(nu ne interesează)
```

Scanned with CamScanner

```
push AX ; punem rezultatul impartirii anterioare pe
stiva deoarece avem nev. de AX pt. conversie
mov AL, [a]; AL = [a]
cbw ; AL -> AX = [a]
mov BX ,[b]; BX= [b]
imul BX; DX:AX = [a]*[b]
mov BX, [C]; punem partea low a lui [C] pe BX
mov CX, [C+2]; punem partea high a lui [c] pe CX
sub AX, BX
sbb DX,CX ; efectuăm scăderea din op. a*b-c tratând
                carry cu borrow => în DX:AX = a*b-c
pop BX ; scoatem de pe stiva (d-110)/(a+b-3)
add AX,BX;
adc DX, 0 ; facem operatia (d-110)/(a+b-3)+a*b-c
push DX
push AX; punem rez. pe stiva
pop EAX; salvăm rez. final în EAX




push dword 0
call [exit]
```