q1:

Code1:

[org 0x0100]

xor ax, ax

xor bx,bx

xor cx,cx

xor dx,dx

jmp main

*Sum:*

mov cx, 7

*tag1:*

add ax,[data + bx]

add bx,2

loop tag1

ret

ret

*main:*

call Sum

mov [result],ax

*; call Sum*

mov ax, 0x4c00

int 0x21

*data:* dw 1,2,1,1,1,1,1

*result:* dw 0

Code2:

[org 0x0100]

xor ax, ax

xor bx,bx

xor cx,cx

xor dx,dx

jmp main

*Sum:*

push bp

mov bp, sp

mov ax,[bp + 6]

mov dx, [bp + 4] *; second parameter*

add ax, dx

pop bp

ret

ret

*main:*

mov cx,7

*tag:*

mov dx,[data+bx]

add bx,2

push ax

push dx

call Sum

loop tag

mov [result],ax

*; call Sum*

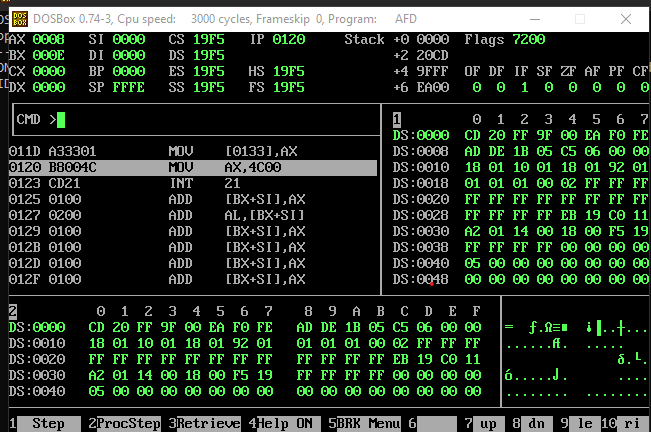
mov ax, 0x4c00

int 0x21

*data:* dw 1,2,1,1,1,1,1

*result:* dw 0

Output:



Q2:

[org 0x0100]

jmp main

*Prod:*

push bp

mov bp, sp

mov ax, [bp + 6] *; frist parameter*

mov bx, [bp + 4] *; second parameter*

mul bx

mov [resultMul],ax

mov ax,[bp + 6]

div bx

mov[resultDiv],ax

pop bp

ret

*main:*

mov ax, 4

mov bx, 3

push ax

push bx

call Prod

mov ax, 0x4c00

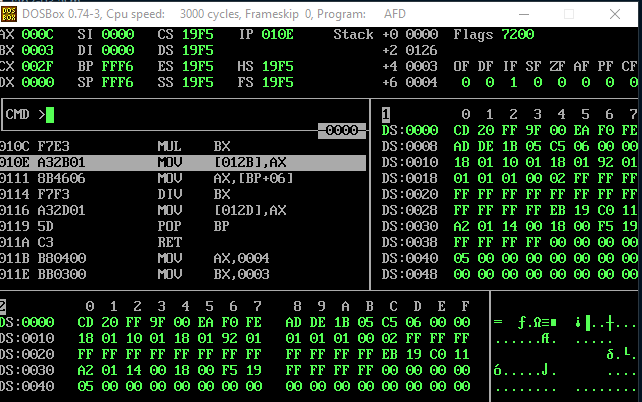
int 0x21

*resultMul:*dw 0

*resultDiv:*dw 0

Output:

Mul;



Div:

