TASK 1:

[org 0x0100]

jmp start

obtained : dw 92

data: dw 0

criteria:

push bp

mov bp,sp

mov cx,[bp+4]

cmp cx,90

mov ax,0x0091

ja result

cmp cx,80

mov ax,0x0081

ja result

cmp cx,70

mov ax,0x0071

ja result

cmp cx,60

mov ax,0x0061

ja result

cmp cx,50

mov ax,0x0051

ja result

cmp cx,40

mov ax,0x0041

ja result

cmp cx,30

mov ax,0x0031

ja result

result:

mov [data],ax

pop bp

pop ax

mov dx,[data]

ret 2

start:

mov ax,[ obtained]

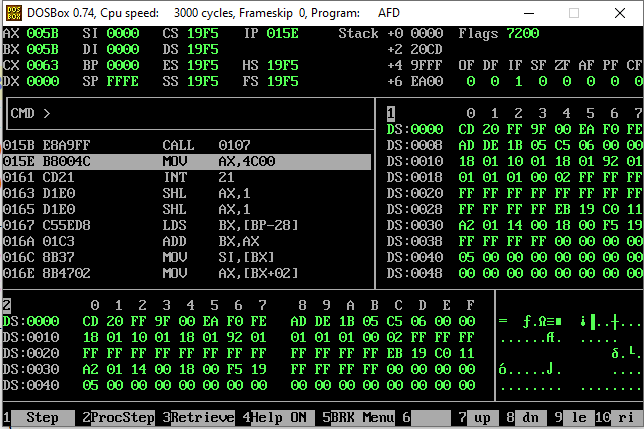
push ax

call criteria

mov ax,0x4c00

int 0x21

## output:



**TASK 2:**

[org 0x0100]

jmp start

data : dw 6

result : dw 0

factorial:

push bp

mov bp,sp

mov ax,[bp+6]

mov bx,[bp+4]

looping:

imul ax,bx

dec bx

cmp bx,1

jne looping

mov [result],ax

pop bp

ret 4

start:

mov bx,[data]

push bx

mov cx,5

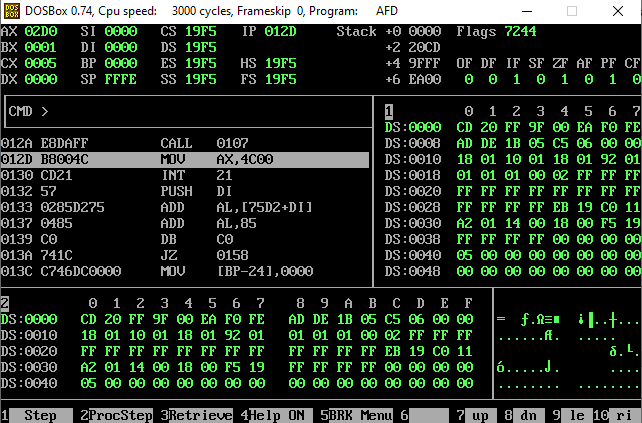
push cx

call factorial

mov ax,0x4c00

int 0x21

## Output;



**Task 3:**

[org 0x0100]

jmp start

data :dw 8

data1:dw 8

data2: dw 0

result:

mov ax,1

mov [data2],ax

pop bp

ret 4

greater:

push bp

mov bp,sp

mov bx,[bp+6]

mov cx,[bp+4]

cmp bx,cx

ja result

pop bp

ret 4

check:

push bp

mov bp,sp

mov dx,[bp+6]

mov cx,[bp+4]

cmp dx,cx

je result

mov ax,2

mov [data2],ax

pop bp

ret 4

start:

mov bx,[data]

push bx

mov cx,[data1]

push cx

call greater

mov ax,[data2]

mov bx,[data]

push bx

mov cx,[data1]

push cx

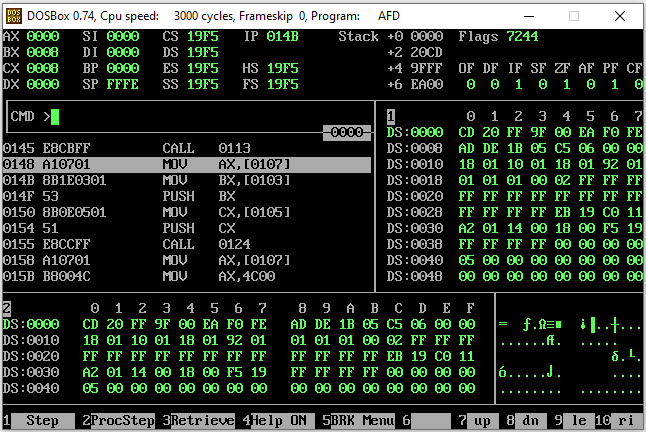
call check

mov ax,[data2]

mov ax,0x4c00

int 0x21

A:



B:

