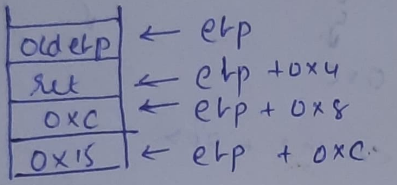
Question - 4.

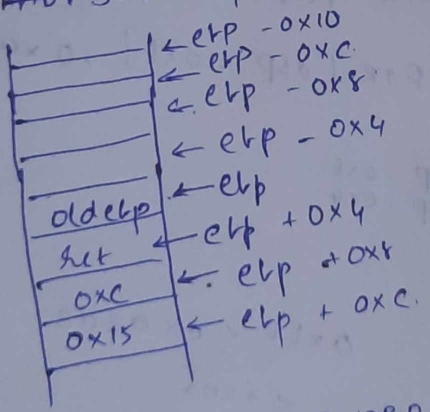
push (0x15)
push (0xC)
call assembly code

<+0>: push ebp # it will push ebp to stack.
<+1>: mov ebp, esp # it moves the content in ebp to esp

After this the status of stack :-



<+3> : ~~mov~~ sub esp ,0x10 #0x10 in hexadecimal ⇒ 16 = 4×4 so space creates for 4 variables.



<+6> : mov $\underbrace{\text{eax}}_{\text{temporary register taken for file.}}$, $\underbrace{\text{DWORD PTR[ebp +0xC]}}$ content in this the address of stack is ~~[illegible]~~ moved to eax.

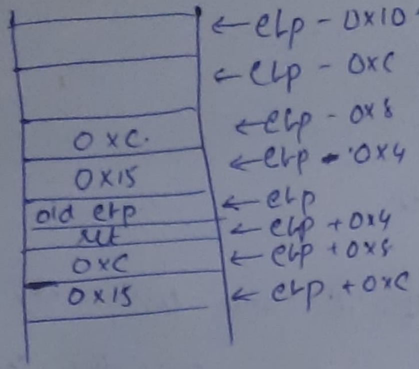
<+9> : mov DWORD PTR[ebp - 0x4] , eax. move the content in eax. Now again to address of stack ebp - 0x4.

lly

<+12> : mov eax , DWORD PTR[ebp + 0x8]

<+15> : mov DWORD PTR.[ebp - 0x8] , eax.
(ebp+0x8) content in this address of stack is moved to eax & then to ebp-0x8 address.

Now the stack status is :-



<+18>: jmp 0x50c <asm2 + 31>

Now unconditionally jump to <+31> in asm2.

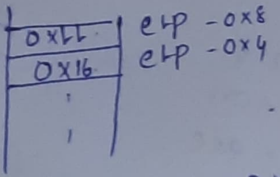
<+31>: cmp DWORD PTR [ebp - 0x8] , 0xa3d3

0xc

<+38> : jle 0x501. <asm2 + 20>.

<+20>: add DWORD PTR [ebp - 0x4] , 0x1

<+24>: add DWORD PTR [ebp - 0x8] , 0xaf



0xbb < 0xa3d3

So it jmp again to <+20>

this process goes on

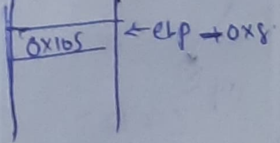
until value in the address ebp - 0x4 is not less than equal to 0xa3d3

so for this to happen 240 times.

loop iterates

0x15 + 240 → 261 = 0x105

21



<+40> : mov eax , DWORD PTR [ebp - 0x4] # now 0x105 pushed to eax.

<+43> : leave # pop the oldframe pointer into Bp & release stack space

<+44> : ret # return the value in eax So, $\boxed{\text{Ans (0x105)}}$