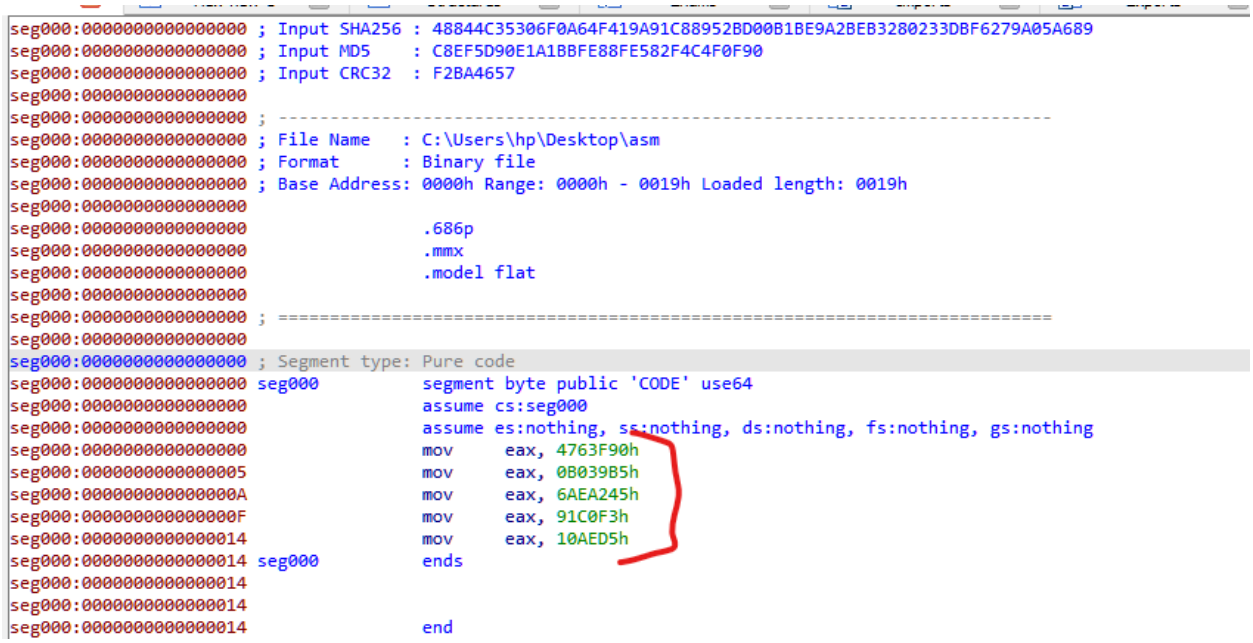
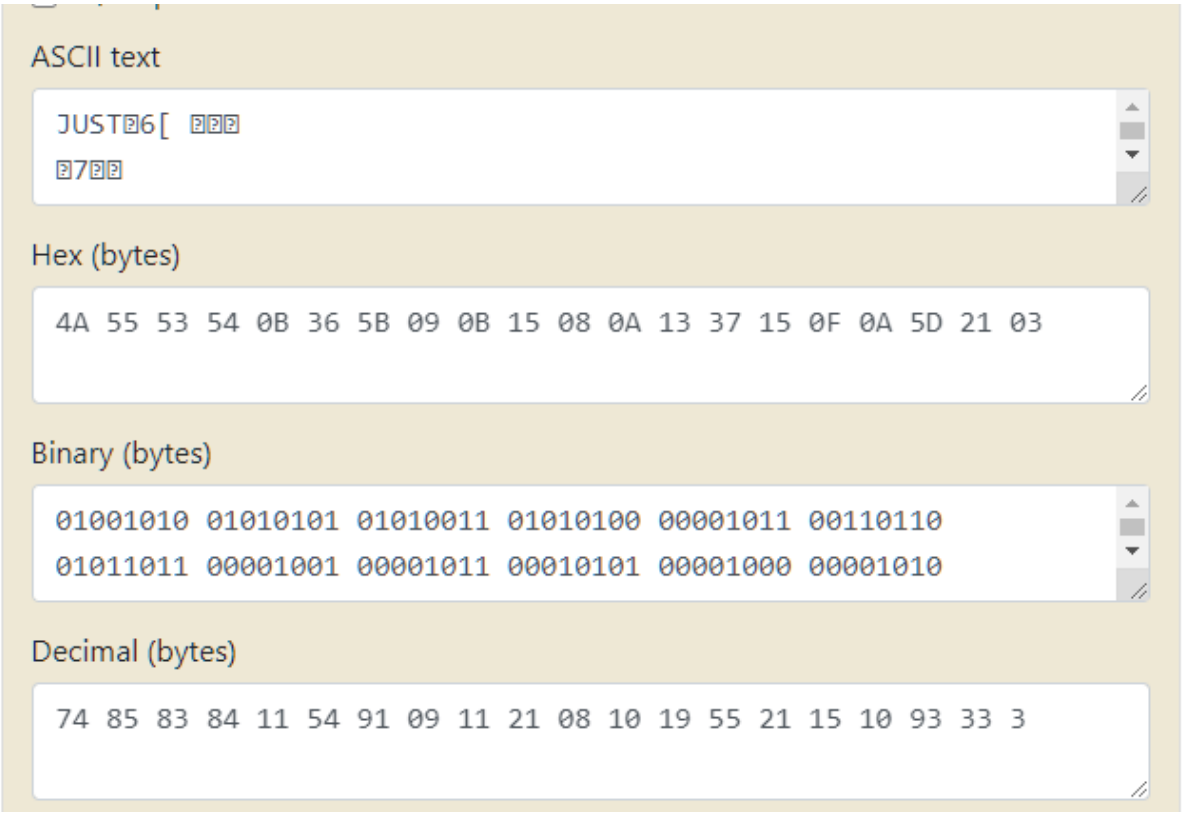
The First and easiest challenge was ASM, all I had to do is to throw the binary file in IDA Pro “Didn’t open In IDA Freeware or in Ghidra and I don’t know why”

We can see that it’s moving hex values to eax



```
seg000:0000000000000000 ; Input SHA256 : 48844C35306F0A64F419A91C88952BD00B1BE9A2BEB3280233DBF6279A05A689
seg000:0000000000000000 ; Input MD5    : C8EF5D90E1A1BBFE88FE582F4C4F0F90
seg000:0000000000000000 ; Input CRC32  : F2BA4657
seg000:0000000000000000
seg000:0000000000000000 ; ---------------------------------------------------------------------------
seg000:0000000000000000 ; File Name   : C:\Users\hp\Desktop\asm
seg000:0000000000000000 ; Format      : Binary file
seg000:0000000000000000 ; Base Address: 0000h Range: 0000h - 0019h Loaded length: 0019h
seg000:0000000000000000
seg000:0000000000000000                 .686p
seg000:0000000000000000                 .mmx
seg000:0000000000000000                 .model flat
seg000:0000000000000000
seg000:0000000000000000 ; ===========================================================================
seg000:0000000000000000
seg000:0000000000000000 ; Segment type: Pure code
seg000:0000000000000000 seg000          segment byte public 'CODE' use64
seg000:0000000000000000                 assume cs:seg000
seg000:0000000000000000                 assume es:nothing, ss:nothing, ds:nothing, fs:nothing, gs:nothing
seg000:0000000000000000                 mov     eax, 4763F90h
seg000:0000000000000005                 mov     eax, 0B039B5h
seg000:000000000000000A                 mov     eax, 6AEA245h
seg000:000000000000000F                 mov     eax, 91C0F3h
seg000:0000000000000014                 mov     eax, 10AED5h
seg000:0000000000000014 seg000          ends
seg000:0000000000000014
seg000:0000000000000014
seg000:0000000000000014                 end
```

All we have to do from here is to take the hex values and convert them to decimals, I did this step online, then we concatenate all the decimal values starting from top to bottom and try to convert them to characters



At this point I knew that my solution is right , but something is still missing so I went to dcode.fr and analyzed the values I found and I got the following result:

Then we click on ASCII Code and we hit Decrypt and there we see our flag: