

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 ; 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 ; Segment type: Pure code
seg000 segment byte public 'CODE' use64
seg000 assume cs:seg000
seg000 assume es:nothing, ss:nothing, ds:nothing, fs:nothing, gs:nothing
seg000 mov eax, 4763F90h
seg000 mov eax, 0B039B5h
seg000 mov eax, 6AEA245h
seg000 mov eax, 91C0F3h
seg000 mov eax, 10AED5h
seg000 ends
seg000 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

ASCII text

```
JUST06[ 000  
0700
```

Hex (bytes)

```
4A 55 53 54 0B 36 5B 09 0B 15 08 0A 13 37 15 0F 0A 5D 21 03
```

Binary (bytes)

```
01001010 01010101 01010011 01010100 00001011 00110110  
01011011 00001001 00001011 00010101 00001000 00001010
```

Decimal (bytes)

```
74 85 83 84 11 54 91 09 11 21 08 10 19 55 21 15 10 93 33 3
```

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:

Search for a tool

★ SEARCH A TOOL ON DCODE BY KEYWORDS:

★ BROWSE THE [FULL DCODE TOOLS' LIST](#)

Results

dCode's analyzer suggests to investigate:

Warning The text has a short length, this can affect the reliability of the results (see FAQ)

↑↓	↑↓
VIC Cipher	■■■■■
Base 36 Cipher	■■■
ASCII Code	■■
Hexadecimal (Base 16)	■
XOR Cipher	■
Circular Bit Shift	■
Octal System (Base 8)	■
LZW Compression	■

Cryptography > Cipher Identifier

ENCRYPTED MESSAGE IDENTIFIER

★ CIPHERTEXT TO RECOGNIZE

★ CLUES/KEYWORDS (IF ANY)

ANALYZE

See also: [Frequency Analysis](#) – [Index of Coincidence](#)

SYMBOLS IDENTIFIER

➤ Go to: [Symbols Cipher List](#)

Answers to Questions (FAQ)

How to decrypt a cipher text?

To decrypt / decipher an encoded message, it is necessary to know the encryption used (or the encoding method, or the implemented cryptographic principle). Without knowing the technique chosen by the sender of the message, it is impossible to decrypt it (or decode it).

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

Results

★ ASCII output limited to printable characters (control chars and non-ASCII characters replaced by ◊)

↑↓	↑↓
DEC /N	JUSTs1mpl_e_4sm!!
BIN 7bit w/◊	
BIN 8bit	î^
OCT /3	âá1AQ◊m◊H0◊
OCT /1-3	<!1◊J◊A-◊i◊◊
OCT /N	<◊◊!1◊J◊A-◊i◊◊

ASCII CONVERTER

★ ASCII CIPHERTEXT (DECIMAL, HEXADECIMAL, ETC.)

★ PRINT RESULT IN HEXADECIMAL ☐

DECRYPT/CONVERT ASCII

See also: [Binary Code](#)

ASCII ENCODER