Introduction to video reverse engineering

Vittorio Giovara

Brussels 2016-01-29
FOSDEM - Open Media

About me

- Libav/FFmpeg developer (~10 decoders)
VideoLAN association member

- First known use of x264 in broadcasting

- Took part to HEVC/H.265 standardization

- Pupil of Kostya

- vittorio.giovara@gmail.com
koda on Freenode IRC

What

- Reverse engineering can be considered a fundamental element of science
- Understand how things work and find rules about their behaviour
- As such it can be applied to anything

What

- ... but let's focus on digital video for now

Theory

- A *video* is a series of *frames*
- *Frames* are data that represent *images*
- They can either be compressed or not
- Data is packed in *some way*

Many many ways

- Lossless or lossy
- There might be a header
- Frames contain RGB(A), YUV, deltas, entropy, slices, inter/intra prediction...
- Luckily many codecs rip each other off (Real, DivX, VP1-9, and many more)

Categories

- Screencast
- Run-length encoding
- Intermediate
- Entropy-based
- Japanese codecs

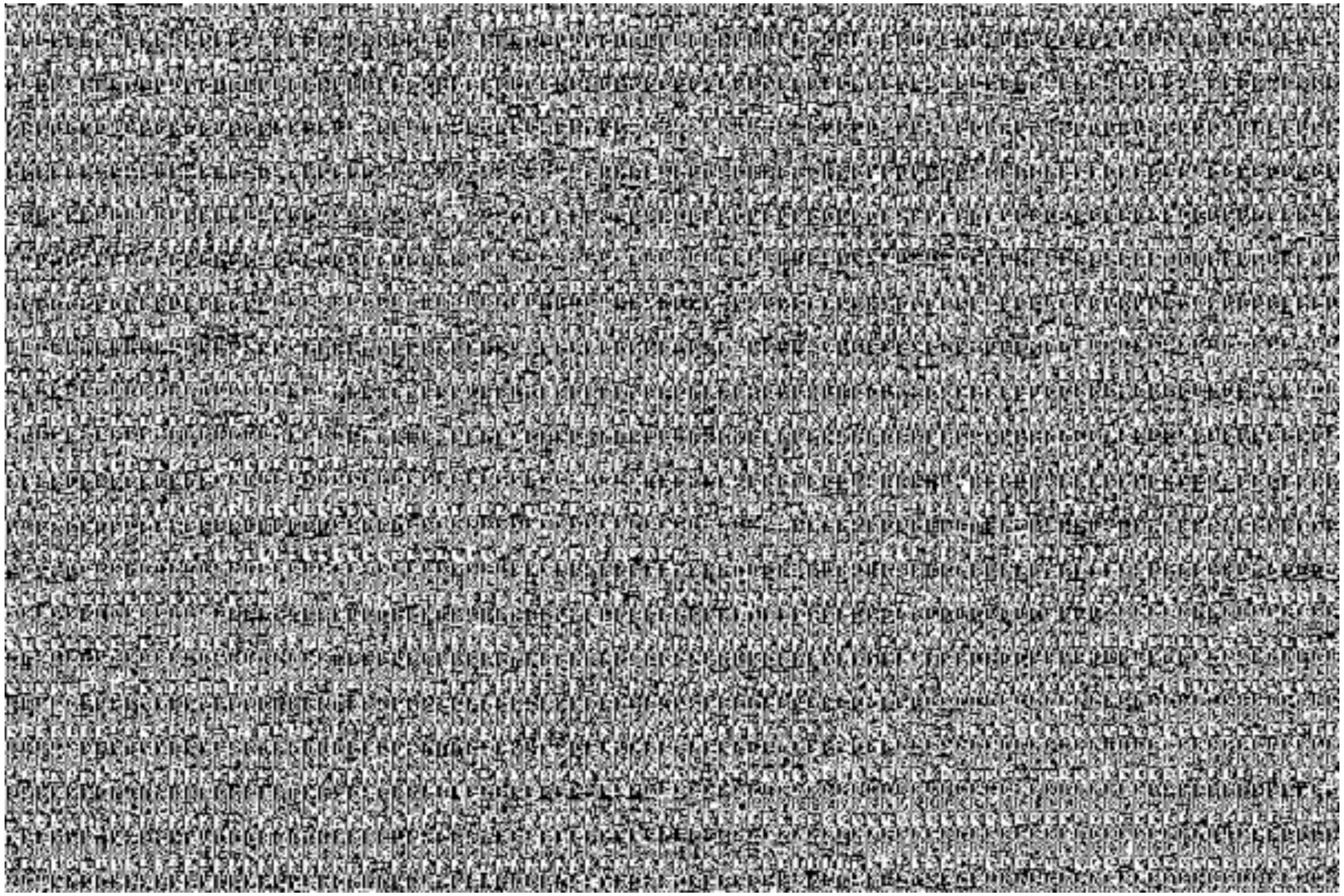
Tools of trade

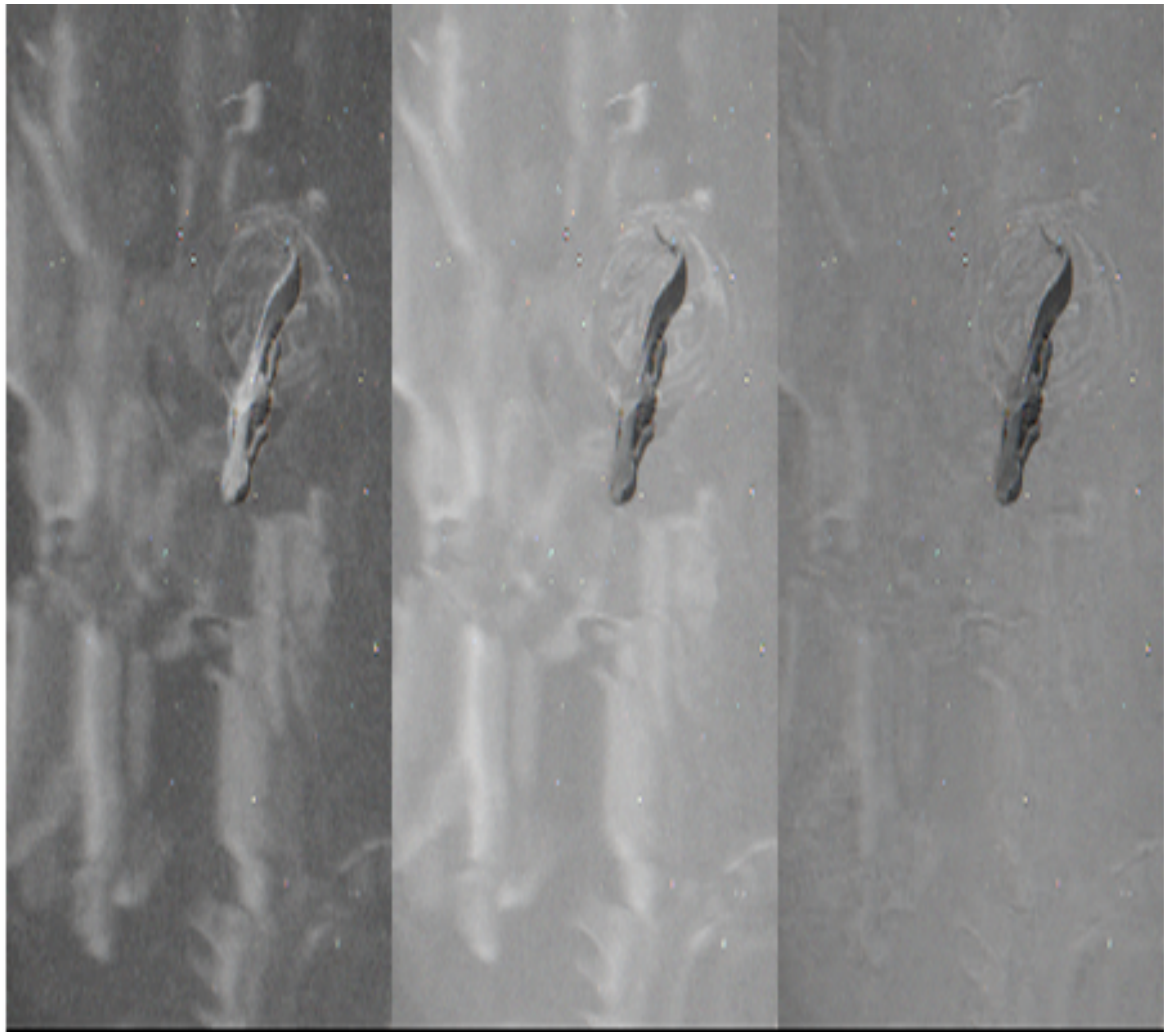
- Common sense
- Specifications and patents
- Strings and debug info
- IDA/HexRays
- Someone to talk with

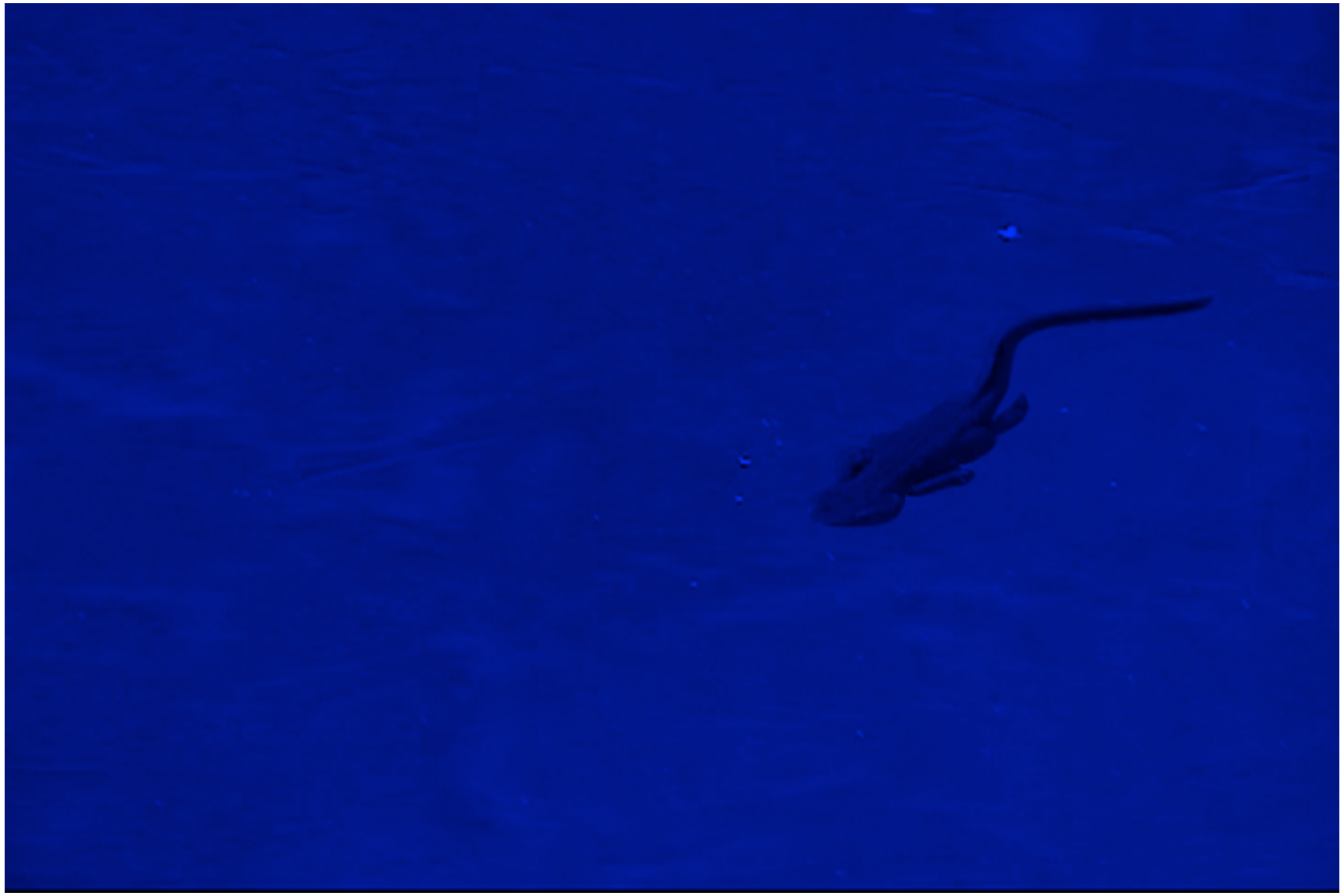
A few examples

- Quickdraw PICT
    - Samples + Spec + Decoder
- TDSC.asf
    - Samples
- CSEUvec.dll
    - Samples + Decoder

PICT





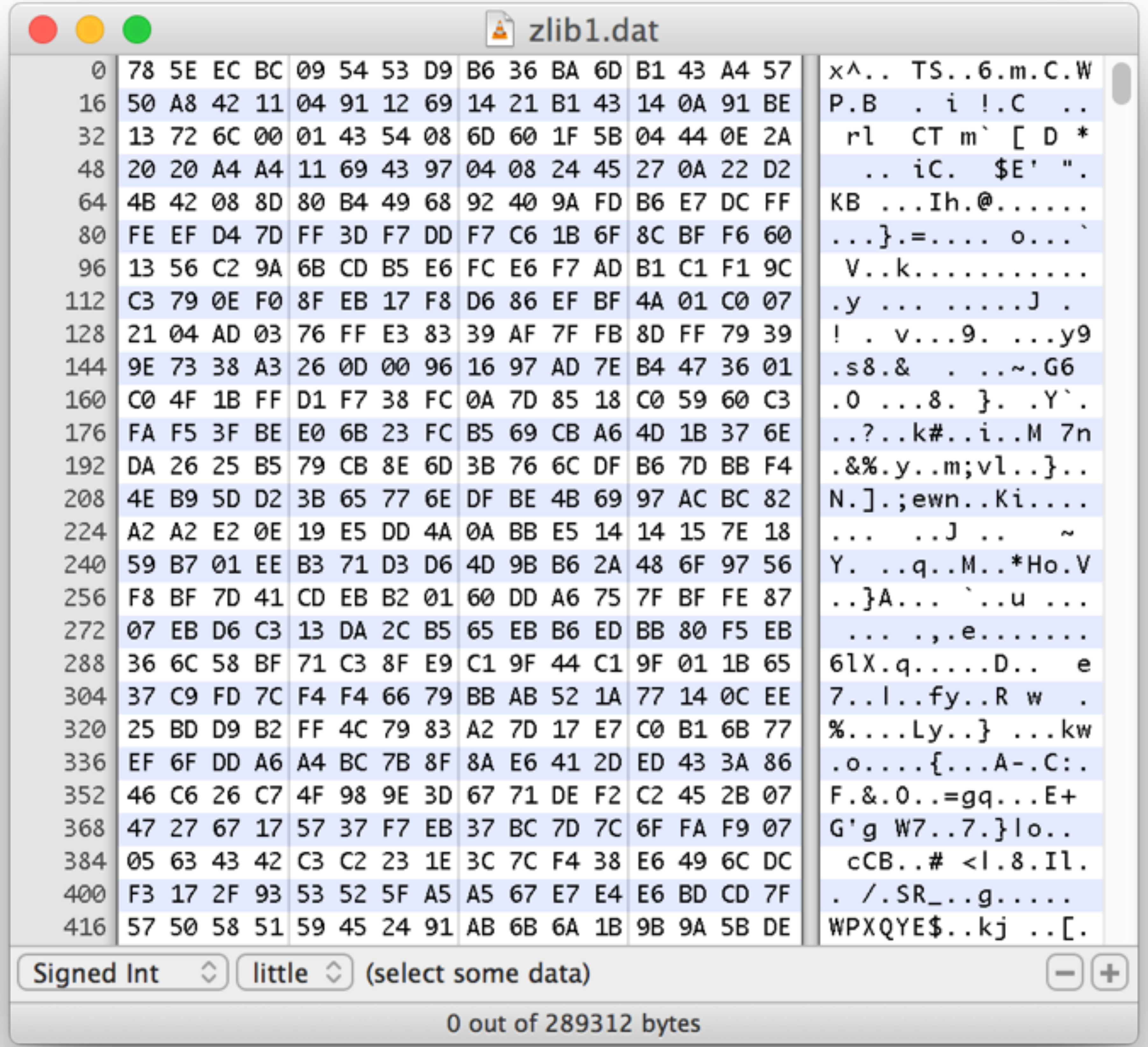


TDSC

```
Format                                    : Windows Media
File size                                 : 39.3 MiB
Duration                                  : 7mn 42s
Overall bit rate mode                     : Variable
Overall bit rate                          : 713 Kbps
Maximum Overall bit rate                  : 717 Kbps
Encoded date                              : UTC 2015-03-02 12:41:49.784

Video
ID                                        : 1
Format                                    : TDSC
Codec ID                                  : TDSC
Bit rate mode                             : Variable
Bit rate                                  : 703 Kbps
Width                                     : 1920 pixels
Height                                    : 1080 pixels
Display aspect ratio                      : 16:9
Frame rate mode                           : Variable
Nominal frame rate                        : 30.000 fps
Bit depth                                 : 8 bits
Language                                  : Chinese (TW)
```

- `./avconv -i ~/tdsc.asf -f image2 -frames 1 zlib1.dat`



zlib1.dat

```
  0  78 5E EC BC 09 54 53 D9 B6 36 BA 6D B1 43 A4 57  x^.. TS..6.m.C.W
 16  50 A8 42 11 04 91 12 69 14 21 B1 43 14 0A 91 BE  P.B  . i !.C  ..
 32  13 72 6C 00 01 43 54 08 6D 60 1F 5B 04 44 0E 2A   rl  CT m` [ D *
 48  20 20 A4 A4 11 69 43 97 04 08 24 45 27 0A 22 D2    .. iC.  $E' ".
 64  4B 42 08 8D 80 B4 49 68 92 40 9A FD B6 E7 DC FF  KB ...Ih.@......
 80  FE EF D4 7D FF 3D F7 DD F7 C6 1B 6F 8C BF F6 60  ...}.=.... o...`
 96  13 56 C2 9A 6B CD B5 E6 FC E6 F7 AD B1 C1 F1 9C   V..k...........
112  C3 79 0E F0 8F EB 17 F8 D6 86 EF BF 4A 01 C0 07  .y ... .....J . 
128  21 04 AD 03 76 FF E3 83 39 AF 7F FB 8D FF 79 39  ! . v...9. ...y9
144  9E 73 38 A3 26 0D 00 96 16 97 AD 7E B4 47 36 01  .s8.&  . ..~.G6 
160  C0 4F 1B FF D1 F7 38 FC 0A 7D 85 18 C0 59 60 C3  .O ...8. }. .Y`.
176  FA F5 3F BE E0 6B 23 FC B5 69 CB A6 4D 1B 37 6E  ..?..k#..i..M 7n
192  DA 26 25 B5 79 CB 8E 6D 3B 76 6C DF B6 7D BB F4  .&%.y..m;vl..}..
208  4E B9 5D D2 3B 65 77 6E DF BE 4B 69 97 AC BC 82  N.].;ewn..Ki....
224  A2 A2 E2 0E 19 E5 DD 4A 0A BB E5 14 14 15 7E 18  ...  ..J ..   ~ 
240  59 B7 01 EE B3 71 D3 D6 4D 9B B6 2A 48 6F 97 56  Y. ..q..M..*Ho.V
256  F8 BF 7D 41 CD EB B2 01 60 DD A6 75 7F BF FE 87  ..}A... `..u ...
272  07 EB D6 C3 13 DA 2C B5 65 EB B6 ED BB 80 F5 EB   ... .,.e.......
288  36 6C 58 BF 71 C3 8F E9 C1 9F 44 C1 9F 01 1B 65  6lX.q.....D..  e
304  37 C9 FD 7C F4 F4 66 79 BB AB 52 1A 77 14 0C EE  7..|..fy..R w  .
320  25 BD D9 B2 FF 4C 79 83 A2 7D 17 E7 C0 B1 6B 77  %....Ly..} ...kw
336  EF 6F DD A6 A4 BC 7B 8F 8A E6 41 2D ED 43 3A 86  .o....{...A-.C:.
352  46 C6 26 C7 4F 98 9E 3D 67 71 DE F2 C2 45 2B 07  F.&.O..=gq...E+ 
368  47 27 67 17 57 37 F7 EB 37 BC 7D 7C 6F FA F9 07  G'g W7..7.}|o.. 
384  05 63 43 42 C3 C2 23 1E 3C 7C F4 38 E6 49 6C DC   cCB..# <|.8.Il.
400  F3 17 2F 93 53 52 5F A5 A5 67 E7 E4 E6 BD CD 7F  . /.SR_..g..... 
416  57 50 58 51 59 45 24 91 AB 6B 6A 1B 9B 9A 5B DE  WPXQYE$..kj ..[.
```

Signed Int ◇ little ◇ (select some data)

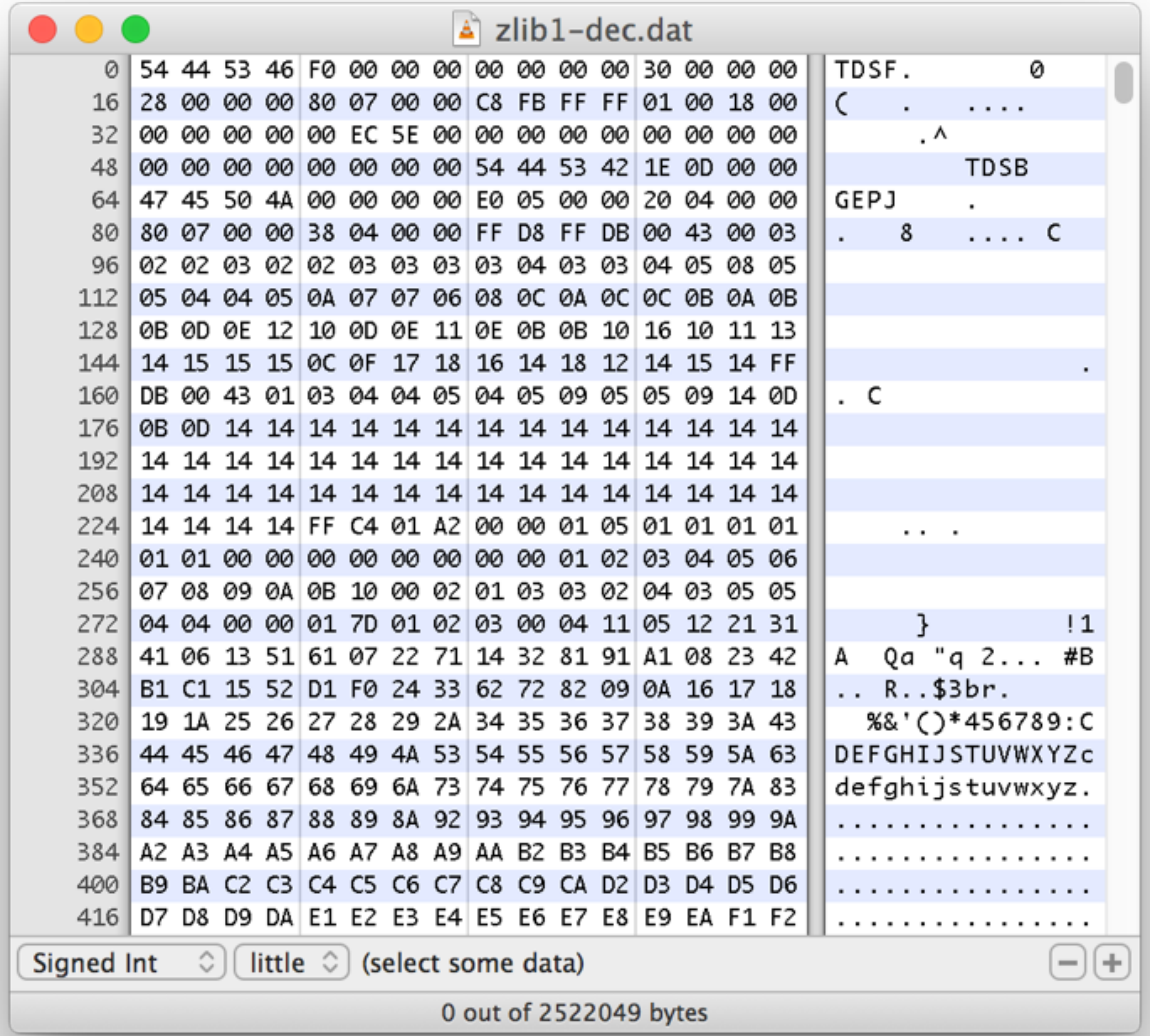
0 out of 289312 bytes

5 line tool

- Try different compressors

```
unsigned char ibuf[SIZE], obuf[SIZE * 10];
int main(void)
{
    uLong ilen, olen;
    ilen = fread(ibuf, 1, sizeof(ibuf), stdin);
    olen = sizeof(obuf);
    uncompress(obuf, &olen, ibuf, ilen);
    fwrite(obuf, 1, olen, stdout);
    return 0;
}
```

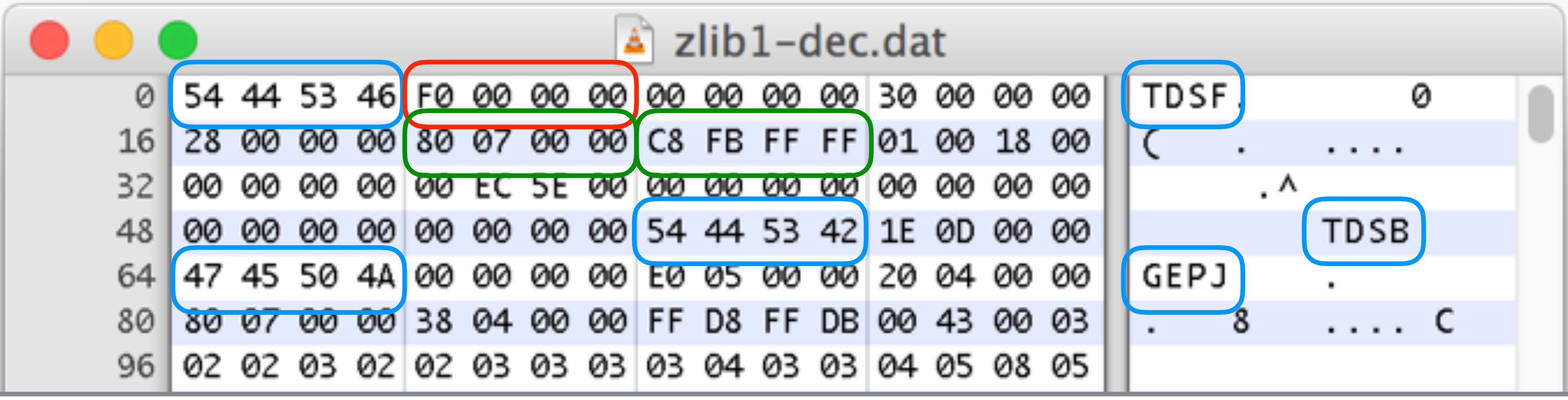
- Can be easily extended to skip the header dynamically



```
  0  54 44 53 46 F0 00 00 00 00 00 00 00 30 00 00 00   TDSF.       0   
 16  28 00 00 00 80 07 00 00 C8 FB FF FF 01 00 18 00   (   .   ....    
 32  00 00 00 00 00 EC 5E 00 00 00 00 00 00 00 00 00        .^         
 48  00 00 00 00 00 00 00 00 54 44 53 42 1E 0D 00 00           TDSB    
 64  47 45 50 4A 00 00 00 00 E0 05 00 00 20 04 00 00   GEPJ    .       
 80  80 07 00 00 38 04 00 00 FF D8 FF DB 00 43 00 03   .   8   .... C  
 96  02 02 03 02 02 03 03 03 03 04 03 03 04 05 08 05                   
112  05 04 04 05 0A 07 07 06 08 0C 0A 0C 0C 0B 0A 0B                   
128  0B 0D 0E 12 10 0D 0E 11 0E 0B 0B 10 16 10 11 13                   
144  14 15 15 15 0C 0F 17 18 16 14 18 12 14 15 14 FF                  .
160  DB 00 43 01 03 04 04 05 04 05 09 05 05 09 14 0D   . C             
176  0B 0D 14 14 14 14 14 14 14 14 14 14 14 14 14 14                   
192  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14                   
208  14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14                   
224  14 14 14 14 FF C4 01 A2 00 00 01 05 01 01 01 01       .. .        
240  01 01 00 00 00 00 00 00 00 00 01 02 03 04 05 06                   
256  07 08 09 0A 0B 10 00 02 01 03 03 02 04 03 05 05                   
272  04 04 00 00 01 7D 01 02 03 00 04 11 05 12 21 31        }        !1
288  41 06 13 51 61 07 22 71 14 32 81 91 A1 08 23 42   A  Qa "q 2... #B
304  B1 C1 15 52 D1 F0 24 33 62 72 82 09 0A 16 17 18   .. R..$3br.     
320  19 1A 25 26 27 28 29 2A 34 35 36 37 38 39 3A 43     %&'()*456789:C
336  44 45 46 47 48 49 4A 53 54 55 56 57 58 59 5A 63   DEFGHIJSTUVWXYZc
352  64 65 66 67 68 69 6A 73 74 75 76 77 78 79 7A 83   defghijstuvwxyz.
368  84 85 86 87 88 89 8A 92 93 94 95 96 97 98 99 9A   ................
384  A2 A3 A4 A5 A6 A7 A8 A9 AA B2 B3 B4 B5 B6 B7 B8   ................
400  B9 BA C2 C3 C4 C5 C6 C7 C8 C9 CA D2 D3 D4 D5 D6   ................
416  D7 D8 D9 DA E1 E2 E3 E4 E5 E6 E7 E8 E9 EA F1 F2   ................
```

Signed Int ⌃ little ⌃ (select some data)

0 out of 2522049 bytes



```
 0  54 44 53 46 F0 00 00 00 00 00 00 00 30 00 00 00   TDSF.          0
16  28 00 00 00 80 07 00 00 C8 FB FF FF 01 00 18 00   (    .    ....
32  00 00 00 00 00 EC 5E 00 00 00 00 00 00 00 00 00        .^
48  00 00 00 00 00 00 00 00 54 44 53 42 1E 0D 00 00           TDSB
64  47 45 50 4A 00 00 00 00 E0 05 00 00 20 04 00 00   GEPJ      .
80  80 07 00 00 38 04 00 00 FF D8 FF DB 00 43 00 03   .   8    .... C
96  02 02 03 02 02 03 03 03 03 04 03 03 04 05 08 05
```

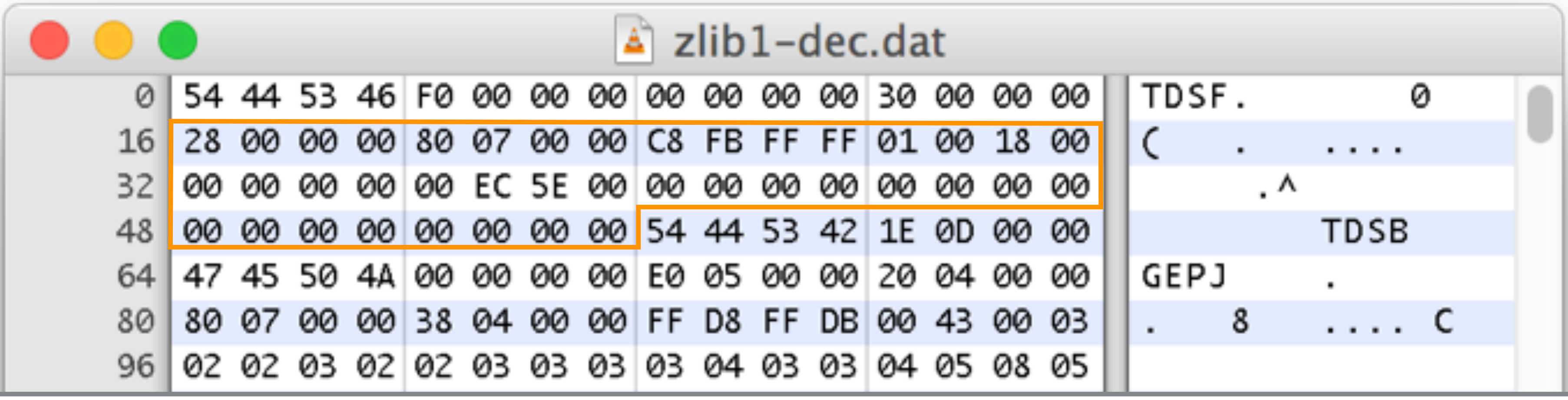
- Tag based

- GEPJ is JPEG in little endian
  later in the file, WAR means RAW

- Count the readable tags, they are 240

- `0x80070000` is 1920
  `0xC8FFFFFF` is -1080

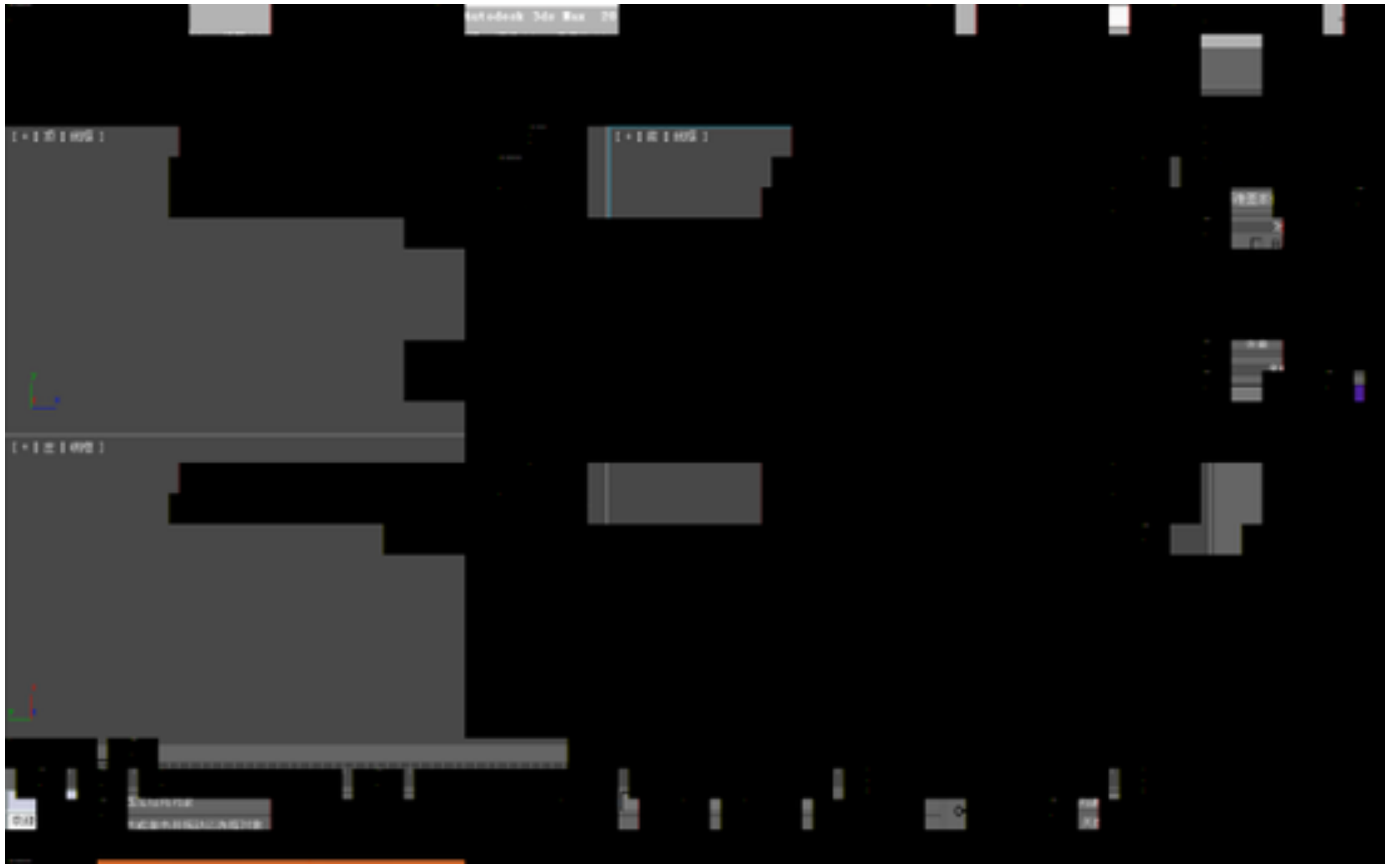
- The `0x28` next to size is suspicious



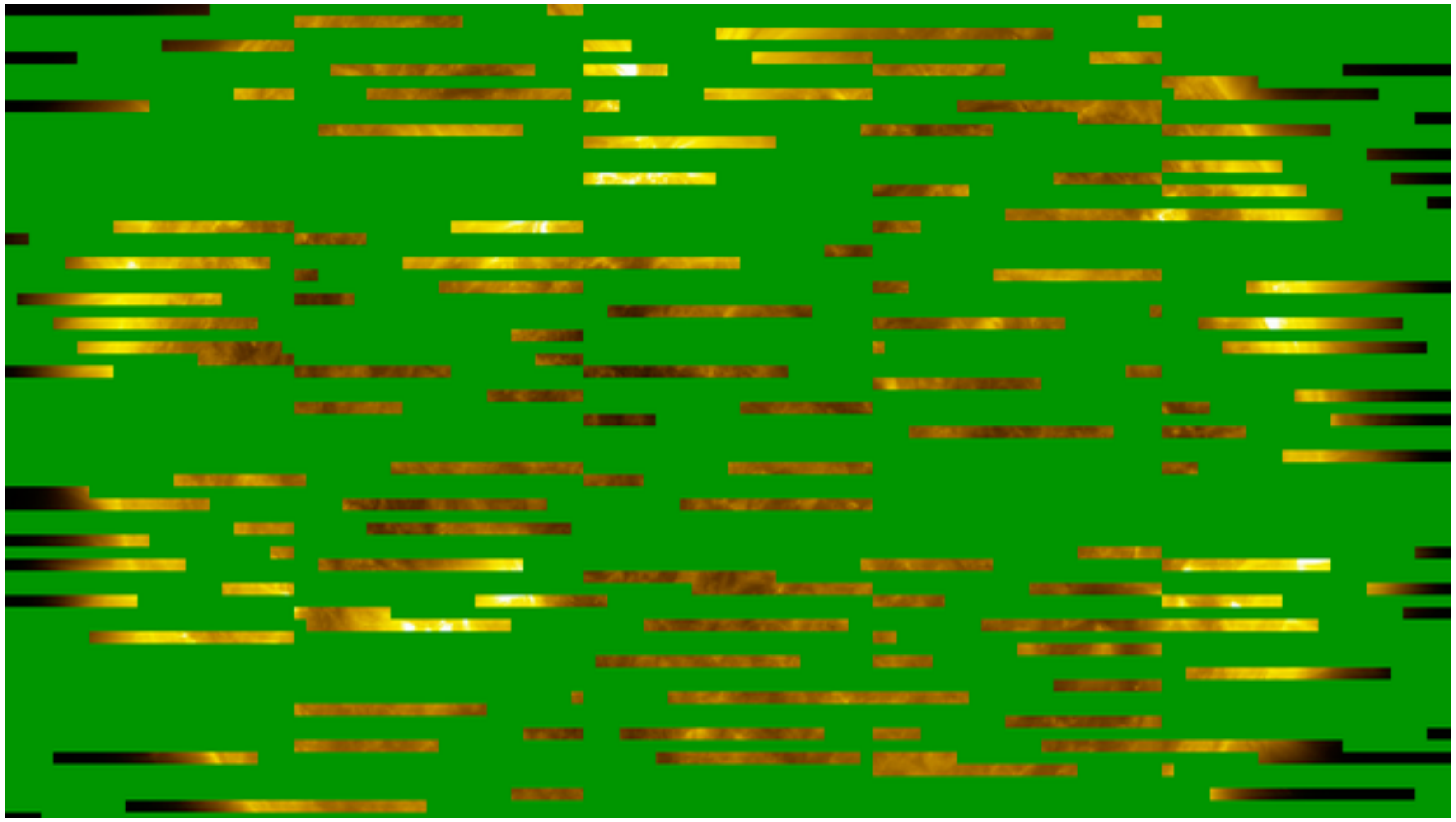
```
 0  54 44 53 46 F0 00 00 00 00 00 00 00 30 00 00 00   TDSF.       0
16  28 00 00 00 80 07 00 00 C8 FB FF FF 01 00 18 00   (   .   ....
32  00 00 00 00 00 EC 5E 00 00 00 00 00 00 00 00 00          .^
48  00 00 00 00 00 00 00 00 54 44 53 42 1E 0D 00 00           TDSB
64  47 45 50 4A 00 00 00 00 E0 05 00 00 20 04 00 00   GEPJ    .
80  80 07 00 00 38 04 00 00 FF D8 FF DB 00 43 00 03   .   8   ....  C
96  02 02 03 02 02 03 03 03 03 04 03 03 04 05 08 05
```

```
typedef struct tagBITMAPINFOHEADER {
  DWORD biSize;
  LONG  biWidth;
  LONG  biHeight;
  WORD  biPlanes;
  WORD  biBitCount;
  DWORD biCompression;
  DWORD biSizeImage;
  LONG  biXPelsPerMeter;
  LONG  biYPelsPerMeter;
  DWORD biClrUsed;
  DWORD biClrImportant;
} BITMAPINFOHEADER;
```

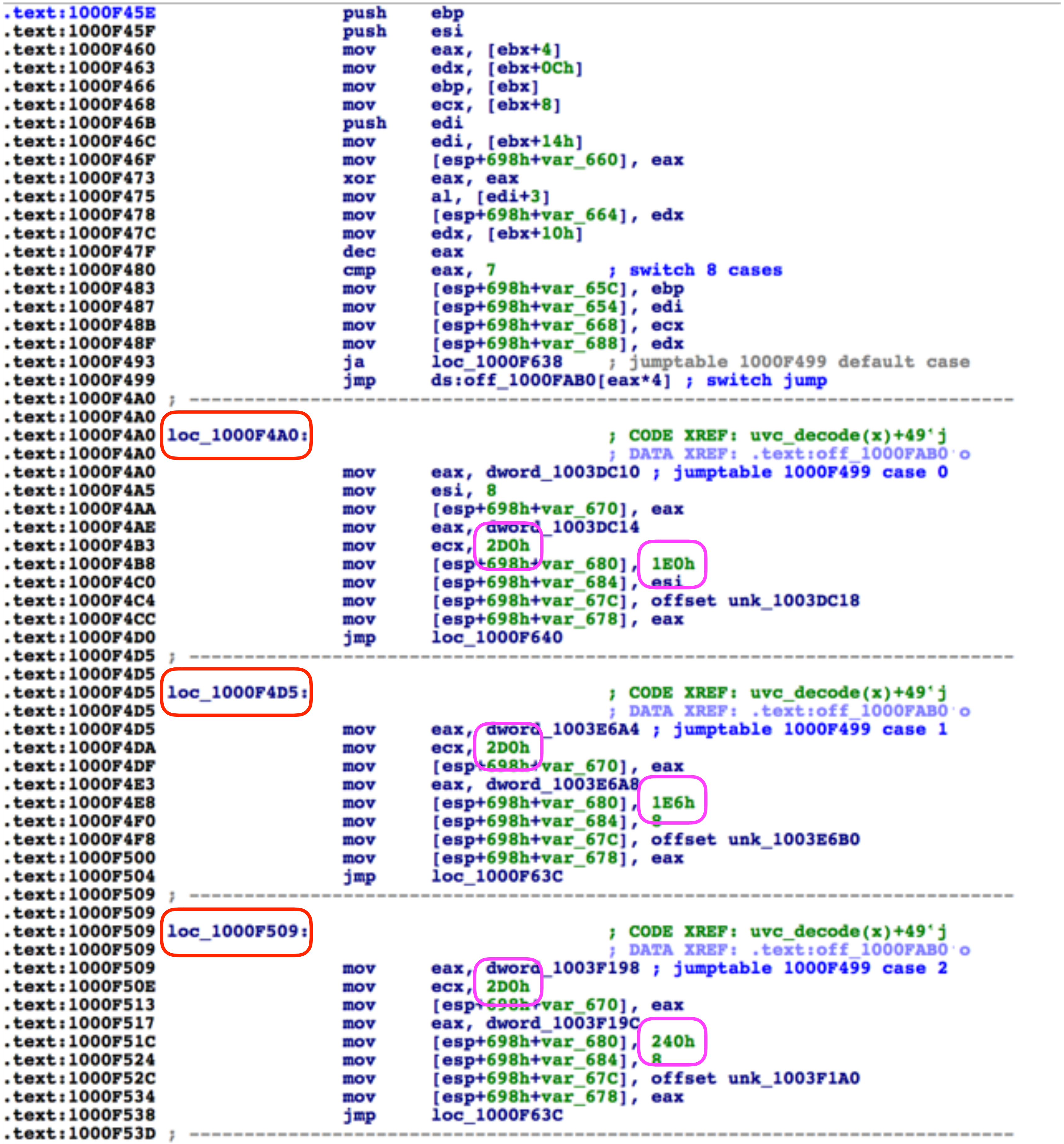
- Every frames is ZLIB-compressed
- TAG-based format with tiles
- Uses Windows-header style
- Has mixed JPEG and RAW data



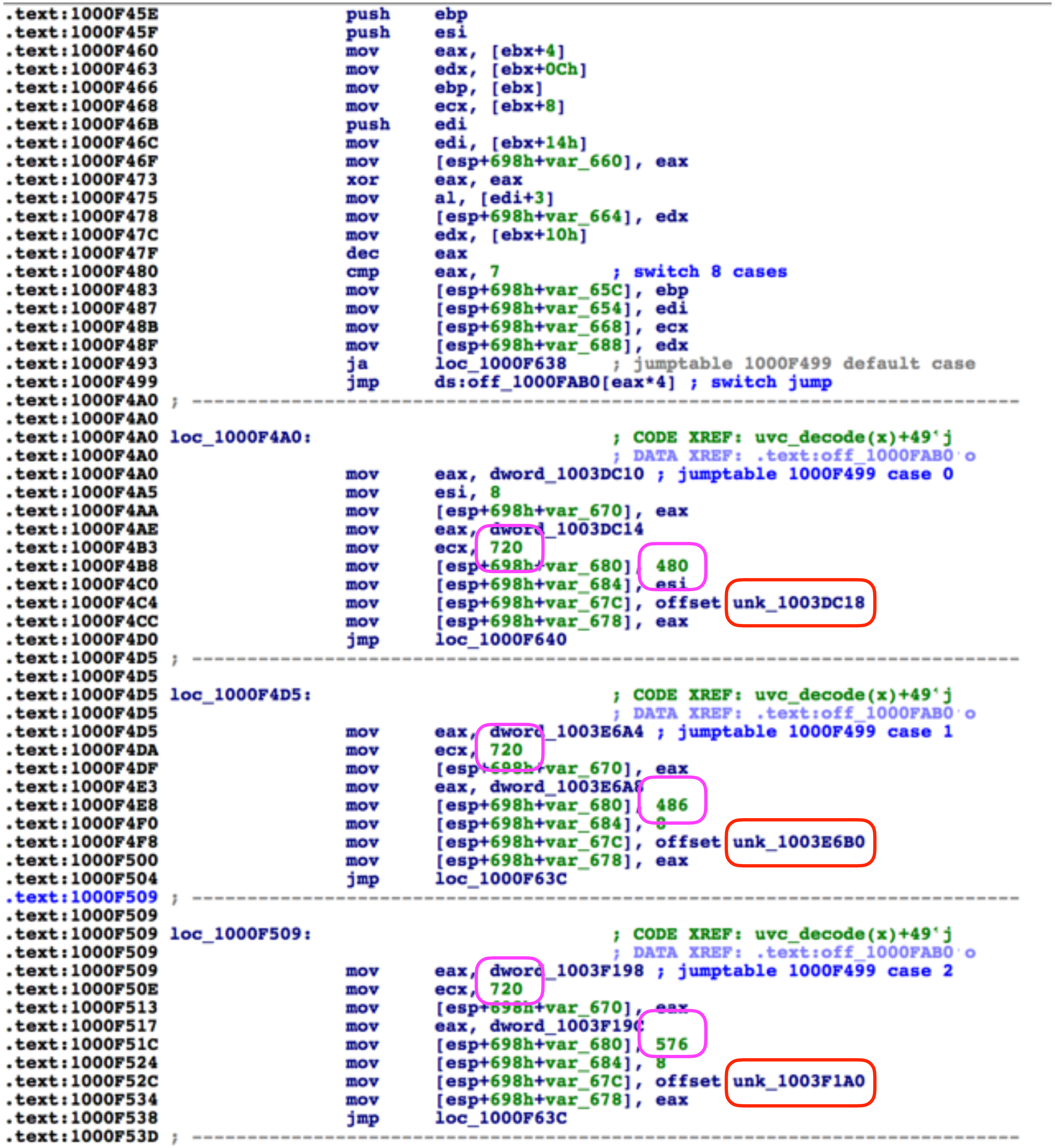
Canopus HQX



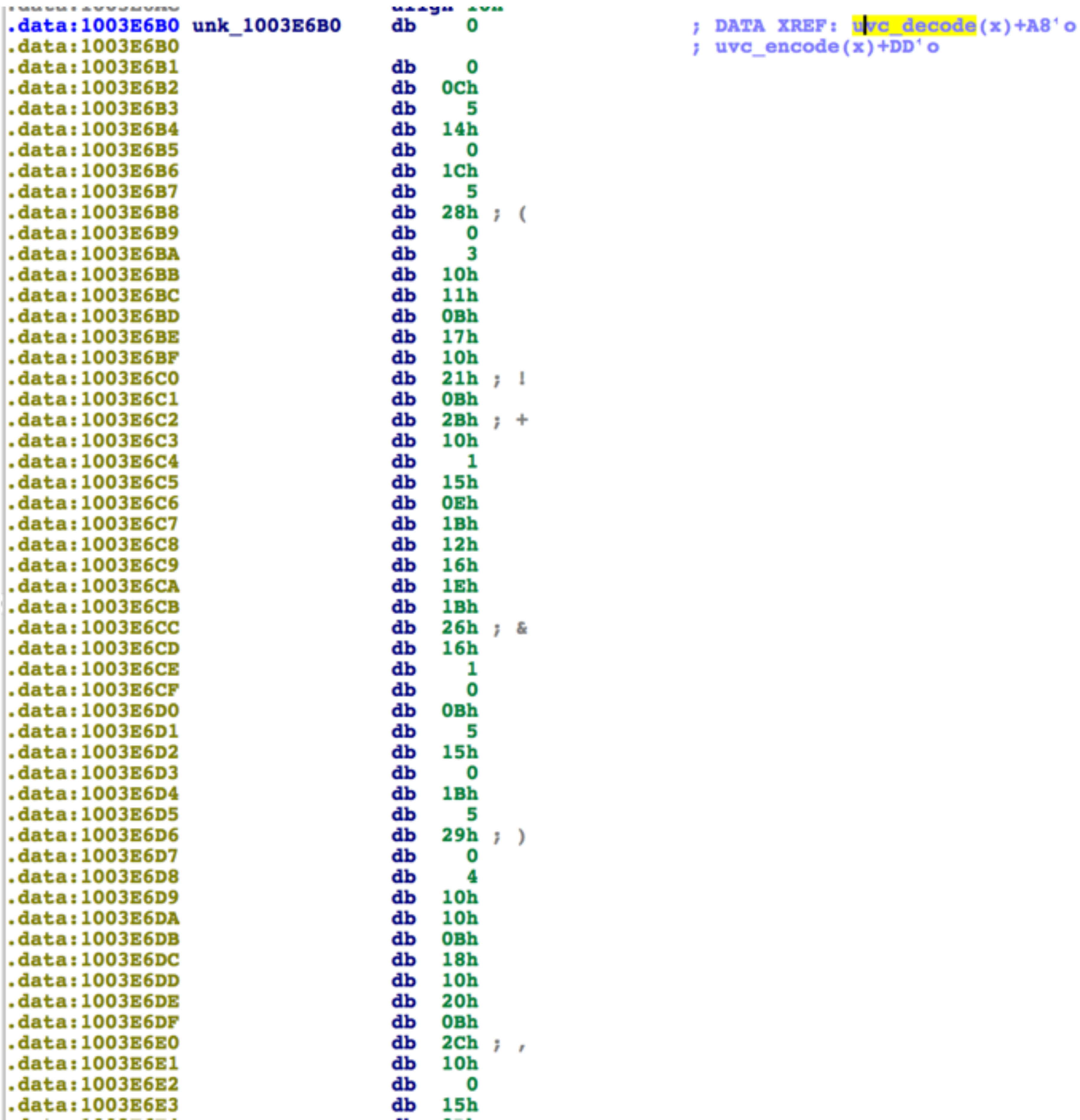
STOP - IDA TIME



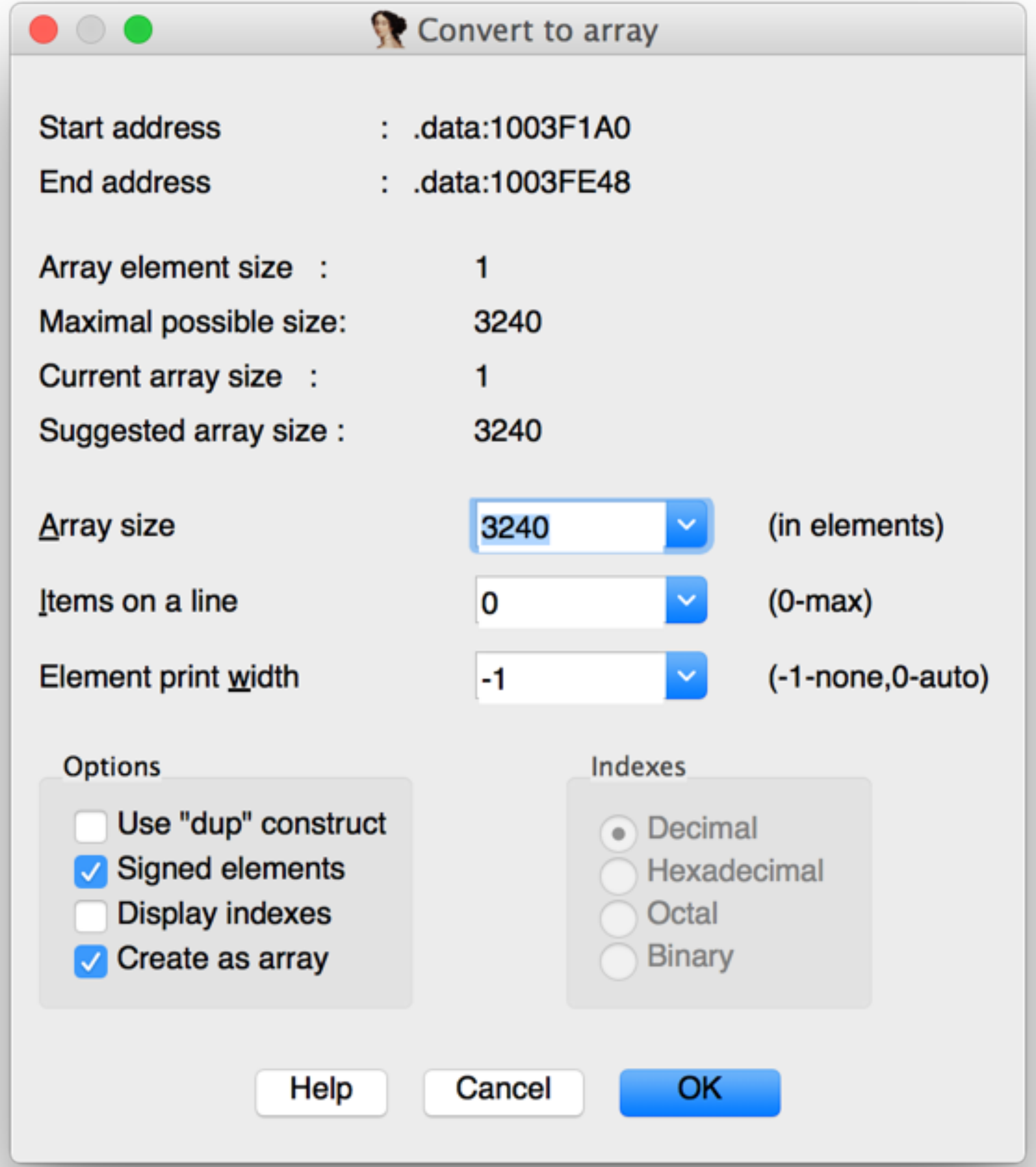
```
.text:1000F45E                 push    ebp
.text:1000F45F                 push    esi
.text:1000F460                 mov     eax, [ebx+4]
.text:1000F463                 mov     edx, [ebx+0Ch]
.text:1000F466                 mov     ebp, [ebx]
.text:1000F468                 mov     ecx, [ebx+8]
.text:1000F46B                 push    edi
.text:1000F46C                 mov     edi, [ebx+14h]
.text:1000F46F                 mov     [esp+698h+var_660], eax
.text:1000F473                 xor     eax, eax
.text:1000F475                 mov     al, [edi+3]
.text:1000F478                 mov     [esp+698h+var_664], edx
.text:1000F47C                 mov     edx, [ebx+10h]
.text:1000F47F                 dec     eax
.text:1000F480                 cmp     eax, 7          ; switch 8 cases
.text:1000F483                 mov     [esp+698h+var_65C], ebp
.text:1000F487                 mov     [esp+698h+var_654], edi
.text:1000F48B                 mov     [esp+698h+var_668], ecx
.text:1000F48F                 mov     [esp+698h+var_688], edx
.text:1000F493                 ja      loc_1000F638    ; jumptable 1000F499 default case
.text:1000F499                 jmp     ds:off_1000FAB0[eax*4] ; switch jump
.text:1000F4A0 ; ---------------------------------------------------------------------------
.text:1000F4A0
.text:1000F4A0 loc_1000F4A0:                           ; CODE XREF: uvc_decode(x)+49'j
.text:1000F4A0                                         ; DATA XREF: .text:off_1000FAB0'o
.text:1000F4A0                 mov     eax, dword_1003DC10 ; jumptable 1000F499 case 0
.text:1000F4A5                 mov     esi, 8
.text:1000F4AA                 mov     [esp+698h+var_670], eax
.text:1000F4AE                 mov     eax, dword_1003DC14
.text:1000F4B3                 mov     ecx, 2D0h
.text:1000F4B8                 mov     [esp+698h+var_680], 1E0h
.text:1000F4C0                 mov     [esp+698h+var_684], esi
.text:1000F4C4                 mov     [esp+698h+var_67C], offset unk_1003DC18
.text:1000F4CC                 mov     [esp+698h+var_678], eax
.text:1000F4D0                 jmp     loc_1000F640
.text:1000F4D5 ; ---------------------------------------------------------------------------
.text:1000F4D5
.text:1000F4D5 loc_1000F4D5:                           ; CODE XREF: uvc_decode(x)+49'j
.text:1000F4D5                                         ; DATA XREF: .text:off_1000FAB0'o
.text:1000F4D5                 mov     eax, dword_1003E6A4 ; jumptable 1000F499 case 1
.text:1000F4DA                 mov     ecx, 2D0h
.text:1000F4DF                 mov     [esp+698h+var_670], eax
.text:1000F4E3                 mov     eax, dword_1003E6A8
.text:1000F4E8                 mov     [esp+698h+var_680], 1E6h
.text:1000F4F0                 mov     [esp+698h+var_684], 8
.text:1000F4F8                 mov     [esp+698h+var_67C], offset unk_1003E6B0
.text:1000F500                 mov     [esp+698h+var_678], eax
.text:1000F504                 jmp     loc_1000F63C
.text:1000F509 ; ---------------------------------------------------------------------------
.text:1000F509
.text:1000F509 loc_1000F509:                           ; CODE XREF: uvc_decode(x)+49'j
.text:1000F509                                         ; DATA XREF: .text:off_1000FAB0'o
.text:1000F509                 mov     eax, dword_1003F198 ; jumptable 1000F499 case 2
.text:1000F50E                 mov     ecx, 2D0h
.text:1000F513                 mov     [esp+698h+var_670], eax
.text:1000F517                 mov     eax, dword_1003F19C
.text:1000F51C                 mov     [esp+698h+var_680], 240h
.text:1000F524                 mov     [esp+698h+var_684], 8
.text:1000F52C                 mov     [esp+698h+var_67C], offset unk_1003F1A0
.text:1000F534                 mov     [esp+698h+var_678], eax
.text:1000F538                 jmp     loc_1000F63C
.text:1000F53D ; ---------------------------------------------------------------------------
```



```
.text:1000F45E                 push    ebp
.text:1000F45F                 push    esi
.text:1000F460                 mov     eax, [ebx+4]
.text:1000F463                 mov     edx, [ebx+0Ch]
.text:1000F466                 mov     ebp, [ebx]
.text:1000F468                 mov     ecx, [ebx+8]
.text:1000F46B                 push    edi
.text:1000F46C                 mov     edi, [ebx+14h]
.text:1000F46F                 mov     [esp+698h+var_660], eax
.text:1000F473                 xor     eax, eax
.text:1000F475                 mov     al, [edi+3]
.text:1000F478                 mov     [esp+698h+var_664], edx
.text:1000F47C                 mov     edx, [ebx+10h]
.text:1000F47F                 dec     eax
.text:1000F480                 cmp     eax, 7          ; switch 8 cases
.text:1000F483                 mov     [esp+698h+var_65C], ebp
.text:1000F487                 mov     [esp+698h+var_654], edi
.text:1000F48B                 mov     [esp+698h+var_668], ecx
.text:1000F48F                 mov     [esp+698h+var_688], edx
.text:1000F493                 ja      loc_1000F638    ; jumptable 1000F499 default case
.text:1000F499                 jmp     ds:off_1000FAB0[eax*4] ; switch jump
.text:1000F4A0 ; ---------------------------------------------------------------------------
.text:1000F4A0
.text:1000F4A0 loc_1000F4A0:                           ; CODE XREF: uvc_decode(x)+49↑j
.text:1000F4A0                                         ; DATA XREF: .text:off_1000FAB0↑o
.text:1000F4A0                 mov     eax, dword_1003DC10 ; jumptable 1000F499 case 0
.text:1000F4A5                 mov     esi, 8
.text:1000F4AA                 mov     [esp+698h+var_670], eax
.text:1000F4AE                 mov     eax, dword_1003DC14
.text:1000F4B3                 mov     ecx, 720
.text:1000F4B8                 mov     [esp+698h+var_680], 480
.text:1000F4C0                 mov     [esp+698h+var_684], esi
.text:1000F4C4                 mov     [esp+698h+var_67C], offset unk_1003DC18
.text:1000F4CC                 mov     [esp+698h+var_678], eax
.text:1000F4D0                 jmp     loc_1000F640
.text:1000F4D5 ; ---------------------------------------------------------------------------
.text:1000F4D5
.text:1000F4D5 loc_1000F4D5:                           ; CODE XREF: uvc_decode(x)+49↑j
.text:1000F4D5                                         ; DATA XREF: .text:off_1000FAB0↑o
.text:1000F4D5                 mov     eax, dword_1003E6A4 ; jumptable 1000F499 case 1
.text:1000F4DA                 mov     ecx, 720
.text:1000F4DF                 mov     [esp+698h+var_670], eax
.text:1000F4E3                 mov     eax, dword_1003E6A8
.text:1000F4E8                 mov     [esp+698h+var_680], 486
.text:1000F4F0                 mov     [esp+698h+var_684], 8
.text:1000F4F8                 mov     [esp+698h+var_67C], offset unk_1003E6B0
.text:1000F500                 mov     [esp+698h+var_678], eax
.text:1000F504                 jmp     loc_1000F63C
.text:1000F509 ; ---------------------------------------------------------------------------
.text:1000F509
.text:1000F509 loc_1000F509:                           ; CODE XREF: uvc_decode(x)+49↑j
.text:1000F509                                         ; DATA XREF: .text:off_1000FAB0↑o
.text:1000F509                 mov     eax, dword_1003F198 ; jumptable 1000F499 case 2
.text:1000F50E                 mov     ecx, 720
.text:1000F513                 mov     [esp+698h+var_670], eax
.text:1000F517                 mov     eax, dword_1003F19C
.text:1000F51C                 mov     [esp+698h+var_680], 576
.text:1000F524                 mov     [esp+698h+var_684], 8
.text:1000F52C                 mov     [esp+698h+var_67C], offset unk_1003F1A0
.text:1000F534                 mov     [esp+698h+var_678], eax
.text:1000F538                 jmp     loc_1000F63C
.text:1000F53D ; ---------------------------------------------------------------------------
```



```
.data:1003E6AC                  align 10h
.data:1003E6B0 unk_1003E6B0     db    0                 ; DATA XREF: uvc_decode(x)+A8↑o
.data:1003E6B0                                          ; uvc_encode(x)+DD↑o
.data:1003E6B1                  db    0
.data:1003E6B2                  db  0Ch
.data:1003E6B3                  db    5
.data:1003E6B4                  db  14h
.data:1003E6B5                  db    0
.data:1003E6B6                  db  1Ch
.data:1003E6B7                  db    5
.data:1003E6B8                  db  28h ; (
.data:1003E6B9                  db    0
.data:1003E6BA                  db    3
.data:1003E6BB                  db  10h
.data:1003E6BC                  db  11h
.data:1003E6BD                  db  0Bh
.data:1003E6BE                  db  17h
.data:1003E6BF                  db  10h
.data:1003E6C0                  db  21h ; !
.data:1003E6C1                  db  0Bh
.data:1003E6C2                  db  2Bh ; +
.data:1003E6C3                  db  10h
.data:1003E6C4                  db    1
.data:1003E6C5                  db  15h
.data:1003E6C6                  db  0Eh
.data:1003E6C7                  db  1Bh
.data:1003E6C8                  db  12h
.data:1003E6C9                  db  16h
.data:1003E6CA                  db  1Eh
.data:1003E6CB                  db  1Bh
.data:1003E6CC                  db  26h ; &
.data:1003E6CD                  db  16h
.data:1003E6CE                  db    1
.data:1003E6CF                  db    0
.data:1003E6D0                  db  0Bh
.data:1003E6D1                  db    5
.data:1003E6D2                  db  15h
.data:1003E6D3                  db    0
.data:1003E6D4                  db  1Bh
.data:1003E6D5                  db    5
.data:1003E6D6                  db  29h ; )
.data:1003E6D7                  db    0
.data:1003E6D8                  db    4
.data:1003E6D9                  db  10h
.data:1003E6DA                  db  10h
.data:1003E6DB                  db  0Bh
.data:1003E6DC                  db  18h
.data:1003E6DD                  db  10h
.data:1003E6DE                  db  20h
.data:1003E6DF                  db  0Bh
.data:1003E6E0                  db  2Ch ; ,
.data:1003E6E1                  db  10h
.data:1003E6E2                  db    0
.data:1003E6E3                  db  15h
```



Convert to array

Start address : .data:1003F1A0

End address : .data:1003FE48

Array element size : 1

Maximal possible size: 3240

Current array size : 1

Suggested array size : 3240

Array size 3240 (in elements)

Items on a line 0 (0-max)

Element print width -1 (-1-none,0-auto)

Options

☐ Use "dup" construct

☑ Signed elements

☐ Display indexes

☑ Create as array

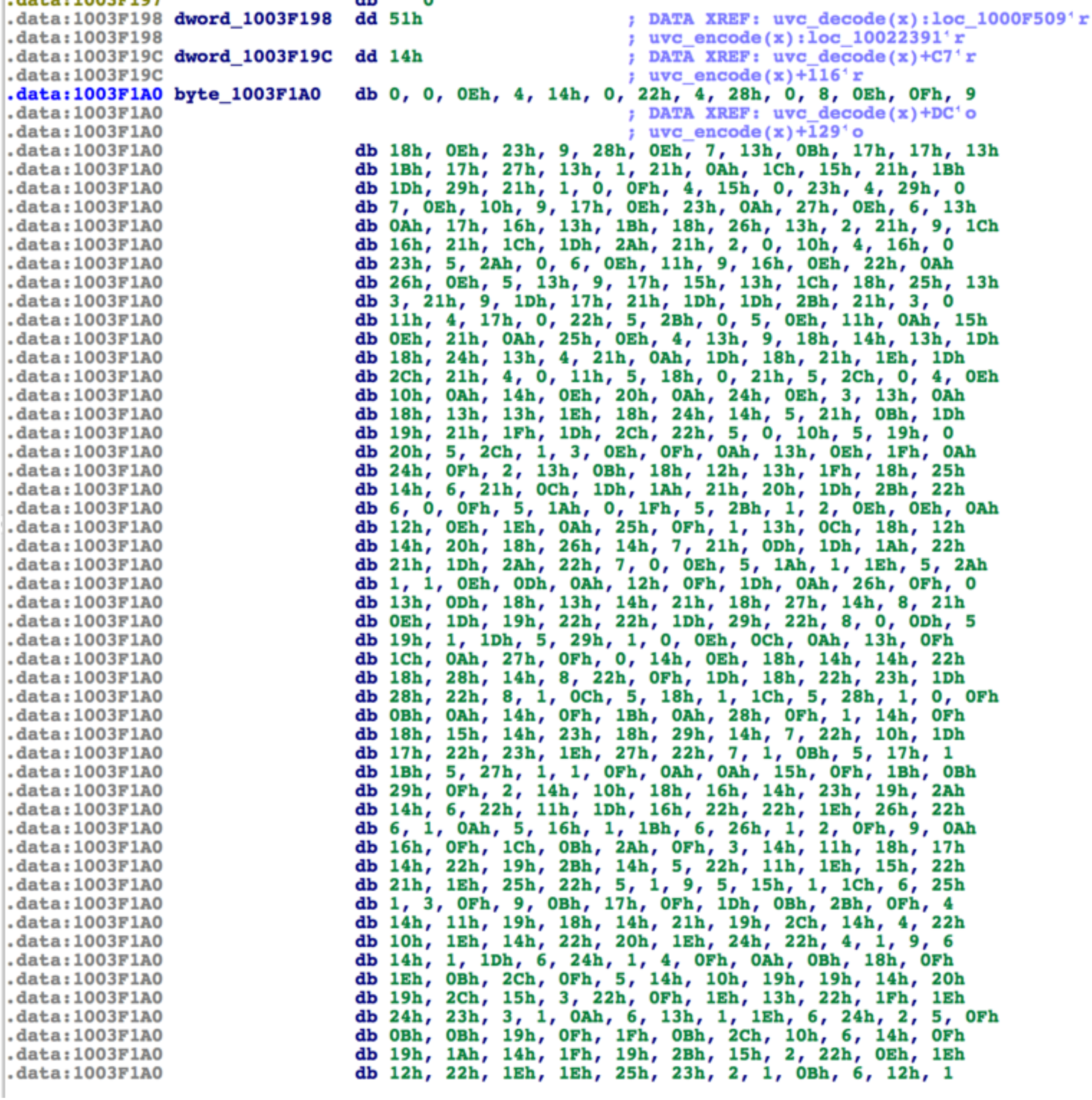
Indexes

● Decimal

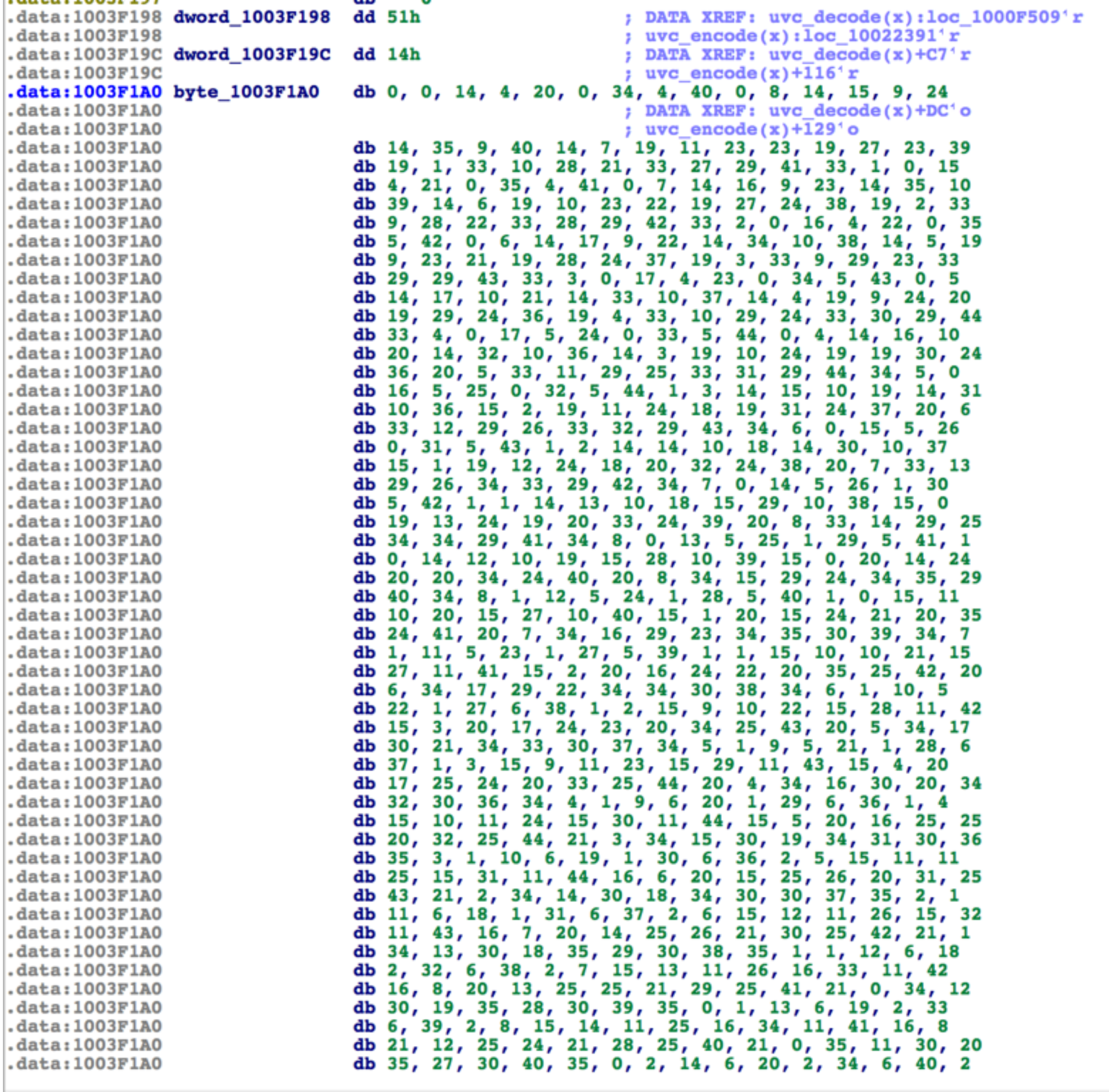
○ Hexadecimal

○ Octal

○ Binary



```
.data:1003F197                          db    0
.data:1003F198 dword_1003F198  dd 51h                   ; DATA XREF: uvc_decode(x):loc_1000F509↑r
.data:1003F198                                          ; uvc_encode(x):loc_10022391↑r
.data:1003F19C dword_1003F19C  dd 14h                   ; DATA XREF: uvc_decode(x)+C7↑r
.data:1003F19C                                          ; uvc_encode(x)+116↑r
.data:1003F1A0 byte_1003F1A0   db 0, 0, 0Eh, 4, 14h, 0, 22h, 4, 28h, 0, 8, 0Eh, 0Fh, 9
.data:1003F1A0                                          ; DATA XREF: uvc_decode(x)+DC↑o
.data:1003F1A0                                          ; uvc_encode(x)+129↑o
.data:1003F1A0                 db 18h, 0Eh, 23h, 9, 28h, 0Eh, 7, 13h, 0Bh, 17h, 17h, 13h
.data:1003F1A0                 db 1Bh, 17h, 27h, 13h, 1, 21h, 0Ah, 1Ch, 15h, 21h, 1Bh
.data:1003F1A0                 db 1Dh, 29h, 21h, 1, 0, 0Fh, 4, 15h, 0, 23h, 4, 29h, 0
.data:1003F1A0                 db 7, 0Eh, 10h, 9, 17h, 0Eh, 23h, 0Ah, 27h, 0Eh, 6, 13h
.data:1003F1A0                 db 0Ah, 17h, 16h, 13h, 1Bh, 18h, 26h, 13h, 2, 21h, 9, 1Ch
.data:1003F1A0                 db 16h, 21h, 1Ch, 1Dh, 2Ah, 21h, 2, 0, 10h, 4, 16h, 0
.data:1003F1A0                 db 23h, 5, 2Ah, 0, 6, 0Eh, 11h, 9, 16h, 0Eh, 22h, 0Ah
.data:1003F1A0                 db 26h, 0Eh, 5, 13h, 9, 17h, 15h, 13h, 1Ch, 18h, 25h, 13h
.data:1003F1A0                 db 3, 21h, 9, 1Dh, 17h, 21h, 1Dh, 1Dh, 2Bh, 21h, 3, 0
.data:1003F1A0                 db 11h, 4, 17h, 0, 22h, 5, 2Bh, 0, 5, 0Eh, 11h, 0Ah, 15h
.data:1003F1A0                 db 0Eh, 21h, 0Ah, 25h, 0Eh, 4, 13h, 9, 18h, 14h, 13h, 1Dh
.data:1003F1A0                 db 18h, 24h, 13h, 4, 21h, 0Ah, 1Dh, 18h, 21h, 1Eh, 1Dh
.data:1003F1A0                 db 2Ch, 21h, 4, 0, 11h, 5, 18h, 0, 21h, 5, 2Ch, 0, 4, 0Eh
.data:1003F1A0                 db 10h, 0Ah, 14h, 0Eh, 20h, 0Ah, 24h, 0Eh, 3, 13h, 0Ah
.data:1003F1A0                 db 18h, 13h, 13h, 1Eh, 18h, 24h, 14h, 5, 21h, 0Bh, 1Dh
.data:1003F1A0                 db 19h, 21h, 1Fh, 1Dh, 2Ch, 22h, 5, 0, 10h, 5, 19h, 0
.data:1003F1A0                 db 20h, 5, 2Ch, 1, 3, 0Eh, 0Fh, 0Ah, 13h, 0Eh, 1Fh, 0Ah
.data:1003F1A0                 db 24h, 0Fh, 2, 13h, 0Bh, 18h, 12h, 13h, 1Fh, 18h, 25h
.data:1003F1A0                 db 14h, 6, 21h, 0Ch, 1Dh, 1Ah, 21h, 20h, 1Dh, 2Bh, 22h
.data:1003F1A0                 db 6, 0, 0Fh, 5, 1Ah, 0, 1Fh, 5, 2Bh, 1, 2, 0Eh, 0Eh, 0Ah
.data:1003F1A0                 db 12h, 0Eh, 1Eh, 0Ah, 25h, 0Fh, 1, 13h, 0Ch, 18h, 12h
.data:1003F1A0                 db 14h, 20h, 18h, 26h, 14h, 7, 21h, 0Dh, 1Dh, 1Ah, 22h
.data:1003F1A0                 db 21h, 1Dh, 2Ah, 22h, 7, 0, 0Eh, 5, 1Ah, 1, 1Eh, 5, 2Ah
.data:1003F1A0                 db 1, 1, 0Eh, 0Dh, 0Ah, 12h, 0Fh, 1Dh, 0Ah, 26h, 0Fh, 0
.data:1003F1A0                 db 13h, 0Dh, 18h, 13h, 14h, 21h, 18h, 27h, 14h, 8, 21h
.data:1003F1A0                 db 0Eh, 1Dh, 19h, 22h, 22h, 1Dh, 29h, 22h, 8, 0, 0Dh, 5
.data:1003F1A0                 db 19h, 1, 1Dh, 5, 29h, 1, 0, 0Eh, 0Ch, 0Ah, 13h, 0Fh
.data:1003F1A0                 db 1Ch, 0Ah, 27h, 0Fh, 0, 14h, 0Eh, 18h, 14h, 14h, 22h
.data:1003F1A0                 db 18h, 28h, 14h, 8, 22h, 0Fh, 1Dh, 18h, 22h, 23h, 1Dh
.data:1003F1A0                 db 28h, 22h, 8, 1, 0Ch, 5, 18h, 1, 1Ch, 5, 28h, 1, 0, 0Fh
.data:1003F1A0                 db 0Bh, 0Ah, 14h, 0Fh, 1Bh, 0Ah, 28h, 0Fh, 1, 14h, 0Fh
.data:1003F1A0                 db 18h, 15h, 14h, 23h, 18h, 29h, 14h, 7, 22h, 10h, 1Dh
.data:1003F1A0                 db 17h, 22h, 23h, 1Eh, 27h, 22h, 7, 1, 0Bh, 5, 17h, 1
.data:1003F1A0                 db 1Bh, 5, 27h, 1, 1, 0Fh, 0Ah, 0Ah, 15h, 0Fh, 1Bh, 0Bh
.data:1003F1A0                 db 29h, 0Fh, 2, 14h, 10h, 18h, 16h, 14h, 23h, 19h, 2Ah
.data:1003F1A0                 db 14h, 6, 22h, 11h, 1Dh, 16h, 22h, 22h, 1Eh, 26h, 22h
.data:1003F1A0                 db 6, 1, 0Ah, 5, 16h, 1, 1Bh, 6, 26h, 1, 2, 0Fh, 9, 0Ah
.data:1003F1A0                 db 16h, 0Fh, 1Ch, 0Bh, 2Ah, 0Fh, 3, 14h, 11h, 18h, 17h
.data:1003F1A0                 db 14h, 22h, 19h, 2Bh, 14h, 5, 22h, 11h, 1Eh, 15h, 22h
.data:1003F1A0                 db 21h, 1Eh, 25h, 22h, 5, 1, 9, 5, 15h, 1, 1Ch, 6, 25h
.data:1003F1A0                 db 1, 3, 0Fh, 9, 0Bh, 17h, 0Fh, 1Dh, 0Bh, 2Bh, 0Fh, 4
.data:1003F1A0                 db 14h, 11h, 19h, 18h, 14h, 21h, 19h, 2Ch, 14h, 4, 22h
.data:1003F1A0                 db 10h, 1Eh, 14h, 22h, 20h, 1Eh, 24h, 22h, 4, 1, 9, 6
.data:1003F1A0                 db 14h, 1, 1Dh, 6, 24h, 1, 4, 0Fh, 0Ah, 0Bh, 18h, 0Fh
.data:1003F1A0                 db 1Eh, 0Bh, 2Ch, 0Fh, 5, 14h, 10h, 19h, 19h, 14h, 20h
.data:1003F1A0                 db 19h, 2Ch, 15h, 3, 22h, 0Fh, 1Eh, 13h, 22h, 1Fh, 1Eh
.data:1003F1A0                 db 24h, 23h, 3, 1, 0Ah, 6, 13h, 1, 1Eh, 6, 24h, 2, 5, 0Fh
.data:1003F1A0                 db 0Bh, 0Bh, 19h, 0Fh, 1Fh, 0Bh, 2Ch, 10h, 6, 14h, 0Fh
.data:1003F1A0                 db 19h, 1Ah, 14h, 1Fh, 19h, 2Bh, 15h, 2, 22h, 0Eh, 1Eh
.data:1003F1A0                 db 12h, 22h, 1Eh, 1Eh, 25h, 23h, 2, 1, 0Bh, 6, 12h, 1
```



```
.data:1003F198 dword_1003F198  dd 51h                      ; DATA XREF: uvc_decode(x):loc_1000F509'r
.data:1003F198                                             ; uvc_encode(x):loc_10022391'r
.data:1003F19C dword_1003F19C  dd 14h                      ; DATA XREF: uvc_decode(x)+C7'r
.data:1003F19C                                             ; uvc_encode(x)+116'r
.data:1003F1A0 byte_1003F1A0   db 0, 0, 14, 4, 20, 0, 34, 4, 40, 0, 8, 14, 15, 9, 24
.data:1003F1A0                                             ; DATA XREF: uvc_decode(x)+DC'o
.data:1003F1A0                                             ; uvc_encode(x)+129'o
.data:1003F1A0                 db 14, 35, 9, 40, 14, 7, 19, 11, 23, 23, 19, 27, 23, 39
.data:1003F1A0                 db 19, 1, 33, 10, 28, 21, 33, 27, 29, 41, 33, 1, 0, 15
.data:1003F1A0                 db 4, 21, 0, 35, 4, 41, 0, 7, 14, 16, 9, 23, 14, 35, 10
.data:1003F1A0                 db 39, 14, 6, 19, 10, 23, 22, 19, 27, 24, 38, 19, 2, 33
.data:1003F1A0                 db 9, 28, 22, 33, 28, 29, 42, 33, 2, 0, 16, 4, 22, 0, 35
.data:1003F1A0                 db 5, 42, 0, 6, 14, 17, 9, 22, 14, 34, 10, 38, 14, 5, 19
.data:1003F1A0                 db 9, 23, 21, 19, 28, 24, 37, 19, 3, 33, 9, 29, 23, 33
.data:1003F1A0                 db 29, 29, 43, 33, 3, 0, 17, 4, 23, 0, 34, 5, 43, 0, 5
.data:1003F1A0                 db 14, 17, 10, 21, 14, 33, 10, 37, 14, 4, 19, 9, 24, 20
.data:1003F1A0                 db 19, 29, 24, 36, 19, 4, 33, 10, 29, 24, 33, 30, 29, 44
.data:1003F1A0                 db 33, 4, 0, 17, 5, 24, 0, 33, 5, 44, 0, 4, 14, 16, 10
.data:1003F1A0                 db 20, 14, 32, 10, 36, 14, 3, 19, 10, 24, 19, 19, 30, 24
.data:1003F1A0                 db 36, 20, 5, 33, 11, 29, 25, 33, 31, 29, 44, 34, 5, 0
.data:1003F1A0                 db 16, 5, 25, 0, 32, 5, 44, 1, 3, 14, 15, 10, 19, 14, 31
.data:1003F1A0                 db 10, 36, 15, 2, 19, 11, 24, 18, 19, 31, 24, 37, 20, 6
.data:1003F1A0                 db 33, 12, 29, 26, 33, 32, 29, 43, 34, 6, 0, 15, 5, 26
.data:1003F1A0                 db 0, 31, 5, 43, 1, 2, 14, 14, 10, 18, 14, 30, 10, 37
.data:1003F1A0                 db 15, 1, 19, 12, 24, 18, 20, 32, 24, 38, 20, 7, 33, 13
.data:1003F1A0                 db 29, 26, 34, 33, 29, 42, 34, 7, 0, 14, 5, 26, 1, 30
.data:1003F1A0                 db 5, 42, 1, 1, 14, 13, 10, 18, 15, 29, 10, 38, 15, 0
.data:1003F1A0                 db 19, 13, 24, 19, 20, 33, 24, 39, 20, 8, 33, 14, 29, 25
.data:1003F1A0                 db 34, 34, 29, 41, 34, 8, 0, 13, 5, 25, 1, 29, 5, 41, 1
.data:1003F1A0                 db 0, 14, 12, 10, 19, 15, 28, 10, 39, 15, 0, 20, 14, 24
.data:1003F1A0                 db 20, 20, 34, 24, 40, 20, 8, 34, 15, 29, 24, 34, 35, 29
.data:1003F1A0                 db 40, 34, 8, 1, 12, 5, 24, 1, 28, 5, 40, 1, 0, 15, 11
.data:1003F1A0                 db 10, 20, 15, 27, 10, 40, 15, 1, 20, 15, 24, 21, 20, 35
.data:1003F1A0                 db 24, 41, 20, 7, 34, 16, 29, 23, 34, 35, 30, 39, 34, 7
.data:1003F1A0                 db 1, 11, 5, 23, 1, 27, 5, 39, 1, 1, 15, 10, 10, 21, 15
.data:1003F1A0                 db 27, 11, 41, 15, 2, 20, 16, 24, 22, 20, 35, 25, 42, 20
.data:1003F1A0                 db 6, 34, 17, 29, 22, 34, 34, 30, 38, 34, 6, 1, 10, 5
.data:1003F1A0                 db 22, 1, 27, 6, 38, 1, 2, 15, 9, 10, 22, 15, 28, 11, 42
.data:1003F1A0                 db 15, 3, 20, 17, 24, 23, 20, 34, 25, 43, 20, 5, 34, 17
.data:1003F1A0                 db 30, 21, 34, 33, 30, 37, 34, 5, 1, 9, 5, 21, 1, 28, 6
.data:1003F1A0                 db 37, 1, 3, 15, 9, 11, 23, 15, 29, 11, 43, 15, 4, 20
.data:1003F1A0                 db 17, 25, 24, 20, 33, 25, 44, 20, 4, 34, 16, 30, 20, 34
.data:1003F1A0                 db 32, 30, 36, 34, 4, 1, 9, 6, 20, 1, 29, 6, 36, 1, 4
.data:1003F1A0                 db 15, 10, 11, 24, 15, 30, 11, 44, 15, 5, 20, 16, 25, 25
.data:1003F1A0                 db 20, 32, 25, 44, 21, 3, 34, 15, 30, 19, 34, 31, 30, 36
.data:1003F1A0                 db 35, 3, 1, 10, 6, 19, 1, 30, 6, 36, 2, 5, 15, 11, 11
.data:1003F1A0                 db 25, 15, 31, 11, 44, 16, 6, 20, 15, 25, 26, 20, 31, 25
.data:1003F1A0                 db 43, 21, 2, 34, 14, 30, 18, 34, 30, 30, 37, 35, 2, 1
.data:1003F1A0                 db 11, 6, 18, 1, 31, 6, 37, 2, 6, 15, 12, 11, 26, 15, 32
.data:1003F1A0                 db 11, 43, 16, 7, 20, 14, 25, 26, 21, 30, 25, 42, 21, 1
.data:1003F1A0                 db 34, 13, 30, 18, 35, 29, 30, 38, 35, 1, 1, 12, 6, 18
.data:1003F1A0                 db 2, 32, 6, 38, 2, 7, 15, 13, 11, 26, 16, 33, 11, 42
.data:1003F1A0                 db 16, 8, 20, 13, 25, 25, 21, 29, 25, 41, 21, 0, 34, 12
.data:1003F1A0                 db 30, 19, 35, 28, 30, 39, 35, 0, 1, 13, 6, 19, 2, 33
.data:1003F1A0                 db 6, 39, 2, 8, 15, 14, 11, 25, 16, 34, 11, 41, 16, 8
.data:1003F1A0                 db 21, 12, 25, 24, 21, 28, 25, 40, 21, 0, 35, 11, 30, 20
.data:1003F1A0                 db 35, 27, 30, 40, 35, 0, 2, 14, 6, 20, 2, 34, 6, 40, 2
```

Why

- You can read the Matrix!
- Avoiding vendor lock-in
  - Cineform/GoPro ♒ SMPTE-VC5
- Fighting digital obsolescence
  - FFV1/MKV archiving codec
- Daala, Thor, VP10 (Open media alliance?)

Thanks

Questions?