Write-up for

NC3 Jule-CTF 2021

opgave "david"

Indledning
Opgaven består er en blok på 201 byte. Nedenfor ses et hexdump af det hele.

```
------------------------------------------------------------------------------
   0   4D 41 50 0A 15 8D D2 5E  33 26 C6 CD A6 70 54 41   MAP....^3&...pTA
  10   42 27 64 6D 68 46 66 2C  E5 6F 72 65 67 7D 3F 55   B'dmhFf,.oreg}?U
  20   6A 61 6E 74 69 73 6B 4D  41 4E E6 79 21 63 33 7B   jantiskMAN.y!c3{
  30   2D 62 75 20 6C 0A 48 2E  76 4D 53 47 F8 EF 7C FA   -bu l.H.vMSG..|.
  40   16 47 81 FF 40 6A DE 3C  61 90 02 3D E3 58 E3 DA   .G..@j.<a..=.X..
  50   5A DB 69 A7 32 22 51 C3  FE 3D F7 DE F7 CF D2 B8   Z.i.2"Q..=......
  60   14 6F B9 EA FE DC C9 D5  2B 7D CF 02 9C 5B 2A 56   .o......+}...[*V
  70   12 8A 43 14 7B BC FD 2B  81 46 FB 9E 84 22 73 6E   ..C.{..+.F..."sn
  80   25 1A A5 66 16 13 AC E3  9A 9F 7B D1 F8 EC 53 91   %..f......{...S.
  90   E1 BE 8E 71 0F C0 DF 5C  20 B3 96 D4 49 3A 69 CE   ...q...\ ...I:i.
  a0   5C 59 62 B9 EF 62 EA EE  F1 B7 6D F0 29 C8 F5 44   \Yb..b....m.)..D
  b0   E7 3A F7 09 46 1B EE 1A  C7 8C 0A 72 62 89 E5 19   .:..F......rb...
  c0   AF 8E FD 8A BA 69 C6 9D  E8                        .....i...
------------------------------------------------------------------------------
```

Filen starter med hvad der ligner et såkaldt magic number 0x4D 0x41 0x50 på offset 0, hvilket bliver til "MAP". Offset 3 (lige efter MAP) indeholder antallet af bytes (0x0a = 10) som ligger før det, der ligner det næste magiske tal (TAB) på offset 0x0E.

Efter TAB står der 0x27 på offset 0x11, hvilket er præcis det antal bytes, som ligger imellem hen til MSG. Efter MSG står der 0xF8, som dog ikke passer med de resterende bytes (ialt 106).

```
------------------------------------------------------------------------------
   0   4D 41 50 0A                                        MAP.
   e   54 41 42 27                                        TAB'
  39   4D 53 47 F8                                        MSG.
------------------------------------------------------------------------------
```

Det ser altså ud til, at filen består af 3 sektioner: MAP med binært indhold, TAB med ASCII-tekst, MSG med binært inhold.

Sektion TAB
TAB betyder sandsynligvis "table" og indholdet ses nedenfor

```
------------------------------------------------------------------------------
  12   64 6D 68 46 66 2C E5 6F  72 65 67 7D 3F 55 6A 61   dmhFf,.oreg}?Uja
  22   6E 74 69 73 6B 4D 41 4E  E6 79 21 63 33 7B 2D 62   ntiskMAN.y!c3{-b
  32   75 20 6C 0A 48 2E 76                               u l.H.v
------------------------------------------------------------------------------
```

Der ses ialt 39 karakterer (inklusiv enkelte danske bogstaver i latin-1). Eftersom denne "tabel" indeholder n, c, 3, og de to klammer { }, er det muligt at skrive et flag med symboler fra denne tabel.

Det bemærkes også at F,U,M,A,N, og H findes i store bogstaver. Det er tænkt som et lille tricky hint i form af "HUFFMAN" (da efternavne ofte staves med stort for at undgå misforståelser).

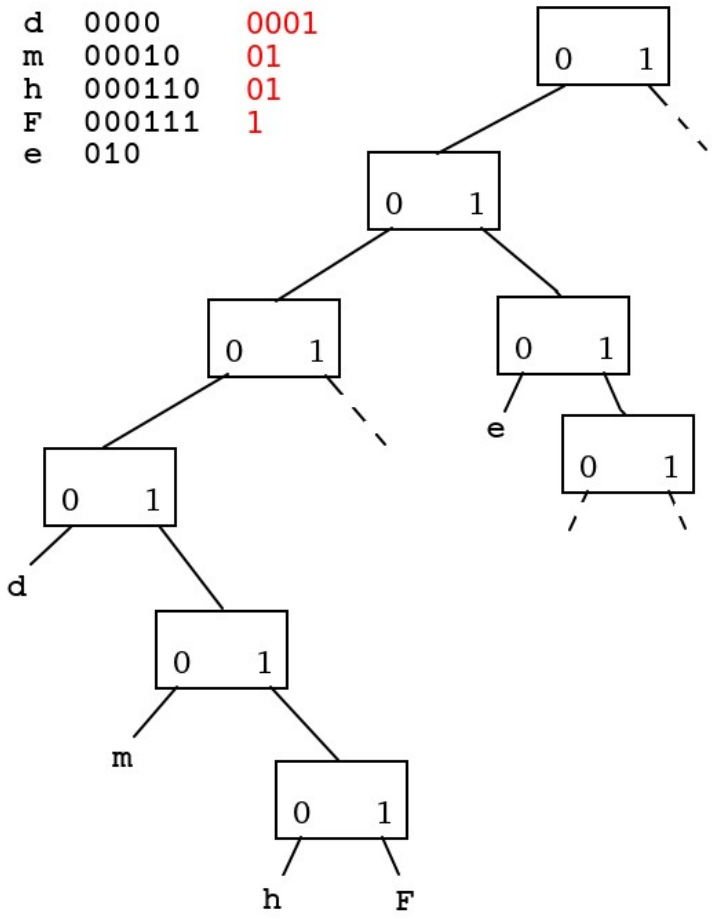
Sektion MAP

Hvis TAB er en symboltabel og MSG er selve beskeden, må det antages at MAP på en aller anden måde kortlægger symbolerne i forhold til det binære indhold i beskeden. Det binære indhold i første sektion ses nedenfor:

```
15 8D D2 5E 33 26 C6 CD A6 70
→ 00010101 10001101 11010010 01011110 00110011 00100110 11000110 11001101 10100110 01110000
```

Det viser sig, at MAP-sektionen repræsenterer et træ, hvor hver node har to børn, som repræsenteres med hver sin bit. Hvis bit'en er '0' er barnet en ny node, og hvis den er '1' er barnet en karakter fra symboltabellen. Dette danner et såkaldt Huffman-træ, som kan bruges til at kode data med varierende bit-længde.

Det ses at antallet af 1-taller (population count) i MAP-sektionen er 39, hvilket er præcis det samme som antallet af karakterer i symboltabellen ovenfor. Dette er ikke nemt at opdage, men ville være et super vigtigt spor i forbindelse med løsning af opgaven. Uddrag af det resulterende træ ses her (bits fra map-sektionen ses i rødt og den resulterende kode i sort):



Da de første 4 bits i MAP er '0001' er det først på den 4. node at det ene barn er en karater, hvilket giver koden '0000' for det første symbol 'd'. Tilsvarende får 'm' den ene halvdel af den næste node, '00010', imens 'h' og 'F' må deles om den næste i form af '000110' og '000111'. Herefter forsætter træet med '001' ... etc, etc.

Nedenstående tabel viser hele dekodningen af de 76 bits fra MAP. I venstre kolonne ses bits'ne og i højre kolonne den resulterende Huffman-kode.

Bit	Char	Kode	Bit	Char	Kode	Bit	Char	Kode
0		0	0		0110101	0		1011100
0		00	1	?	**01101010**	1	y	**10111000**
0		000	1	U	**01101011**	1	!	**10111001**
1	d	**0000**	1	j	**011011**	0		1011101
0		0001	1	a	**0111**	1	c	**10111010**
1	m	**00010**	0		1	1	3	**10111011**
0		00011	0		10	0		101111
1	h	**000110**	0		100	0		1011110
1	F	**000111**	1	n	**1000**	1	{	**10111100**
0		001	1	t	**1001**	1	-	**10111101**
0		0010	0		101	0		1011111
0		00100	0		1010	1	b	**10111110**
1	f	**001000**	1	i	**10100**	1	u	**10111111**
1	,	**001001**	1	s	**10101**	0		11
0		00101	0		1011	1	SPC	**110**
1	å	**001010**	0		10110	0		111
1	o	**001011**	1	k	**101100**	0		1110
1	r	**0011**	0		101101	1	l	**11100**
0		01	0		1011010	1	LF	**11101**
1	e	**010**	1	M	**10110100**	0		1111
0		011	1	A	**10110101**	0		11110
0		0110	0		1011011	1	H	**111100**
1	g	**01100**	1	N	**10110110**	1	.	**111101**
0		01101	1	æ	**10110111**	1	v	**11111**
0		011010	0		10111			
1	}	**0110100**	0		101110			

Dekodning af MSG
Nu da koden er kendt, kan beskeden dekodes bit for bit. De første 9 bytes under MSG ses nedenfor:

```
EF 7C FA 16 47 81 FF 40 6A
```

Dette giver følgende karakterer:

11101	<LF>	0011	r
11101	<LF>	110	<space>
111100	H	0000	d
11111	v	0111	a
010	e	11111	v
00010	m	10100	i
110	<space>	0000	d
010	e	01101010	?

Den samlede besked ender således:

```
------------------------------------------------------------------------
  0   0A 0A 48 76 65 6D 20 65  72 20 64 61 76 69 64 3F   ..Hvem er david?
 10   20 48 61 6E 20 68 65 64  64 65 72 20 48 55 46 46    Han hedder HUFF
 20   4D 41 4E 20 74 69 6C 20  65 66 74 65 72 6E 61 76   MAN til efternav
 30   6E 2E 2E 2E 0A 0A 48 76  69 73 20 64 65 6E 20 76   n.....Hvis den v
 40   61 72 20 73 76 E6 72 2C  20 73 E5 20 76 61 72 20   ar sv.r, s. var
 50   64 65 74 20 6F 67 73 E5  20 6D 65 6E 69 6E 67 65   det ogs. meninge
 60   6E 2E 20 48 76 69 73 20  64 65 6E 20 76 61 72 20   n. Hvis den var
 70   6E 65 6D 2C 20 6A 61 6D  65 6E 20 73 E5 20 67 6F   nem, jamen s. go
 80   64 74 20 6B 6C 61 72 65  74 2E 0A 0A 46 6C 61 67   dt klaret...Flag
 90   65 74 20 65 72 20 68 76  65 72 74 20 66 61 6C 64   et er hvert fald
 a0   20 76 65 6C 66 6F 72 74  6A 65 6E 74 2C 20 74 69    velfortjent, ti
 b0   6C 6C 79 6B 6B 65 21 0A  0A 6E 63 33 7B 6A 61 6A   llykke!..nc3{jaj
 c0   61 20 64 65 74 20 65 72  20 73 6D 61 72 74 20 2D   a det er smart -
 d0   20 6D 65 6E 20 68 76 61  64 20 6B 61 6E 20 64 65    men hvad kan de
 e0   74 20 65 67 65 6E 74 6C  69 67 20 62 72 75 67 65   t egentlig bruge
 f0   73 20 74 69 6C 7D 0A 0A                           s til}..
------------------------------------------------------------------------
```

Forslag til hints

- Hvad hedder David til efternavn? (De eneste bogstaver, der findes med stort er HUFFMAN)
- Filen er inddelt i 3 markerede sektioner med hver deres struktur. (MAP, TAB, MSG)
- Hvad er 'population count' i den første sektion? (Det samme som antallet af symboler)
- Der indeholdes en besked på 248 karakterer som optager ialt 1117 bits. (ca 4.5 bits/karakter)
- Hvor mange bits bruges der til at repræsentere hver karakter? (Det må nødvendigvis variere)

Flaget
Flaget ender med at se således ud:

```
nc3{jaja det er smart - men hvad kan det egentlig bruges til}
```

Bonusinfo
Huffman-kodning er opfundet af David Huffman i 1951.