

1

- $42_8 = \underbrace{00100}_{16} \underbrace{010}_2 = \underbrace{22}_{16} = 2 \cdot 16_{10} + 2_{10} = 34_{10}$
- $255_8 = 10101101_2 = AD_{16} = 173_{10}$
- $3047_8 = 11000100111_2 = 627_{16} = 6 \cdot 16^2 + 32 + 7 = 1575$
- $140336_8 = 1100000111011110_2 = CODE_{16} = 12 \cdot 16^3 + 0 + 13 \cdot 16 + 14 = 49374$

2

- $22_{16} + 8_{16} = 2A_{16}$
- $73_{16} + 2C_{16} = 93_{16} + C_{16} = 9F_{16}$
- $7F_{16} + 7F_{16} = EF_{16} + F_{16} = FE_{16}$
- $C2_{16} + A4_{16} = 162_{16} + 4_{16} = 166_{16}$

7

- 0 (NUL): Pierwotnie NUL był znakiem pustym, tj. ignorowanym, nie wpływający na stan maszyny.
Obecnie w języku C oznacza koniec stringu.
- 4 (EoT): End-of-transmission. Służy do zawiadamiania o rozłączeniu obwodu, odłączeniu terminala, końcu wysyłania wiadomości itd.
- 7 (BEL): Oznacza powiadomienie o nadchodzącej wiadomości (przeważnie za pomocą dźwięku).
- 10 (LF, EOL): End-of-line.
- 12 (FF): Form feed. Oznacza zakończenie obecnej strony i rozpoczęcie następnej (np. dla drukarki)

8

ASCII pozwala na reprezentację jedynie 128 znaków.
UTF-8 pozwala na reprezentację aż do ~miliona symboli.

Przedział	Kodowanie
[0x00, 0x7F]	0xxxxxxx
[0x80, 0x7FF]	110xxxxx 10xxxxxx
[0x800, 0xFFFF]	1110xxxx 10xxxxxx 10xxxxxx
[0x10000, 0x1FFFFF]	11110xxxx 10xxxxxx 10xxxxxx 10xxxxxx

Algorytm rozkodowywania UTF-8:
1) Znajdujemy kod interesującego nas znaku w UTF-8: U+<kod>
2) Znajdujemy odpowiedni zakres
3) Zamieniamy część kodującą <kod> na binarkę
4) Podstawiamy jej kolejne bity pod "x"

Proszę zapłacić 5€! = 50726F737AC499207A6170C582616369C4872035E282AC21