

Informatika 1

3. előadás: Számok és karakterek

Wetttl Ferenc prezentációjának felhasználásával

Budapesti Műszaki és Gazdaságtudományi Egyetem

2017-09-19

Konverzió 2-es számrendszerből 10-es számrendszerbe:

$$b_n b_{n-1} \dots b_1 b_0 . b_{-1} \dots b_{-m} = \sum_{i=-m}^n b_i 2^i.$$

Konverzió 2-es számrendszerből 10-es számrendszerbe:

$$b_n b_{n-1} \dots b_1 b_0 . b_{-1} \dots b_{-m} = \sum_{i=-m}^n b_i 2^i.$$

Például $110.101_2 =$

Konverzió 2-es számrendszerből 10-es számrendszerbe:

$$b_n b_{n-1} \dots b_1 b_0 . b_{-1} \dots b_{-m} = \sum_{i=-m}^n b_i 2^i.$$

Például $110.101_2 = 6.625$

Konverzió 2-es számrendszerből 10-es számrendszerbe:

$$b_n b_{n-1} \dots b_1 b_0 . b_{-1} \dots b_{-m} = \sum_{i=-m}^n b_i 2^i.$$

Például $110.101_2 = 6.625$

Konverzió 10-es számrendszerből 2-es számrendszerbe

- egészek esetén ismételt 2-vel való maradékos osztás,
- törtrész esetén ismételt 2-vel való szorzás.

Konverzió 2-es számrendszerből 10-es számrendszerbe:

$$b_n b_{n-1} \dots b_1 b_0 . b_{-1} \dots b_{-m} = \sum_{i=-m}^n b_i 2^i.$$

Például $110.101_2 = 6.625$

Konverzió 10-es számrendszerből 2-es számrendszerbe

- egészek esetén ismételt 2-vel való maradékos osztás,
- törtrész esetén ismételt 2-vel való szorzás.

Például 106 bináris alakja:

Konverzió 2-es számrendszerből 10-es számrendszerbe:

$$b_n b_{n-1} \dots b_1 b_0 . b_{-1} \dots b_{-m} = \sum_{i=-m}^n b_i 2^i.$$

Például $110.101_2 = 6.625$

Konverzió 10-es számrendszerből 2-es számrendszerbe

- egészek esetén ismételt 2-vel való maradékos osztás,
- törtrész esetén ismételt 2-vel való szorzás.

Például 106 bináris alakja:

$$106 = 2 \cdot 53 + 0$$

Konverzió 2-es számrendszerből 10-es számrendszerbe:

$$b_n b_{n-1} \dots b_1 b_0 . b_{-1} \dots b_{-m} = \sum_{i=-m}^n b_i 2^i.$$

Például $110.101_2 = 6.625$

Konverzió 10-es számrendszerből 2-es számrendszerbe

- egészek esetén ismételt 2-vel való maradékos osztás,
- törtrész esetén ismételt 2-vel való szorzás.

Például 106 bináris alakja:

$$106 = 2 \cdot 53 + 0 \rightarrow 0$$

Konverzió 2-es számrendszerből 10-es számrendszerbe:

$$b_n b_{n-1} \dots b_1 b_0 . b_{-1} \dots b_{-m} = \sum_{i=-m}^n b_i 2^i.$$

Például $110.101_2 = 6.625$

Konverzió 10-es számrendszerből 2-es számrendszerbe

- egészek esetén ismételt 2-vel való maradékos osztás,
- törtrész esetén ismételt 2-vel való szorzás.

Például 106 bináris alakja:

$$106 = 2 \cdot 53 + 0 \rightarrow 0$$

$$53 = 2 \cdot 26 + 1$$

Konverzió 2-es számrendszerből 10-es számrendszerbe:

$$b_n b_{n-1} \dots b_1 b_0 . b_{-1} \dots b_{-m} = \sum_{i=-m}^n b_i 2^i.$$

Például $110.101_2 = 6.625$

Konverzió 10-es számrendszerből 2-es számrendszerbe

- egészek esetén ismételt 2-vel való maradékos osztás,
- törtrész esetén ismételt 2-vel való szorzás.

Például 106 bináris alakja:

$$106 = 2 \cdot 53 + 0 \rightarrow 0$$

$$53 = 2 \cdot 26 + 1 \rightarrow 1$$

Konverzió 2-es számrendszerből 10-es számrendszerbe:

$$b_n b_{n-1} \dots b_1 b_0 . b_{-1} \dots b_{-m} = \sum_{i=-m}^n b_i 2^i.$$

Például $110.101_2 = 6.625$

Konverzió 10-es számrendszerből 2-es számrendszerbe

- egészek esetén ismételt 2-vel való maradékos osztás,
- törtrész esetén ismételt 2-vel való szorzás.

Például 106 bináris alakja:

$$106 = 2 \cdot 53 + 0 \rightarrow 0$$

$$53 = 2 \cdot 26 + 1 \rightarrow 1$$

$$26 = 2 \cdot 13 + 0$$

Konverzió 2-es számrendszerből 10-es számrendszerbe:

$$b_n b_{n-1} \dots b_1 b_0 . b_{-1} \dots b_{-m} = \sum_{i=-m}^n b_i 2^i.$$

Például $110.101_2 = 6.625$

Konverzió 10-es számrendszerből 2-es számrendszerbe

- egészek esetén ismételt 2-vel való maradékos osztás,
- törtrész esetén ismételt 2-vel való szorzás.

Például 106 bináris alakja:

$$106 = 2 \cdot 53 + 0 \rightarrow 0$$

$$53 = 2 \cdot 26 + 1 \rightarrow 1$$

$$26 = 2 \cdot 13 + 0 \rightarrow 0$$

Konverzió 2-es számrendszerből 10-es számrendszerbe:

$$b_n b_{n-1} \dots b_1 b_0 . b_{-1} \dots b_{-m} = \sum_{i=-m}^n b_i 2^i.$$

Például $110.101_2 = 6.625$

Konverzió 10-es számrendszerből 2-es számrendszerbe

- egészek esetén ismételt 2-vel való maradékos osztás,
- törtrész esetén ismételt 2-vel való szorzás.

Például 106 bináris alakja:

$$106 = 2 \cdot 53 + 0 \rightarrow 0$$

$$53 = 2 \cdot 26 + 1 \rightarrow 1$$

$$26 = 2 \cdot 13 + 0 \rightarrow 0$$

$$13 = 2 \cdot 6 + 1$$

Konverzió 2-es számrendszerből 10-es számrendszerbe:

$$b_n b_{n-1} \dots b_1 b_0 . b_{-1} \dots b_{-m} = \sum_{i=-m}^n b_i 2^i.$$

Például $110.101_2 = 6.625$

Konverzió 10-es számrendszerből 2-es számrendszerbe

- egészek esetén ismételt 2-vel való maradékos osztás,
- törtrész esetén ismételt 2-vel való szorzás.

Például 106 bináris alakja:

$$106 = 2 \cdot 53 + 0 \rightarrow 0$$

$$53 = 2 \cdot 26 + 1 \rightarrow 1$$

$$26 = 2 \cdot 13 + 0 \rightarrow 0$$

$$13 = 2 \cdot 6 + 1 \rightarrow 1$$

Konverzió 2-es számrendszerből 10-es számrendszerbe:

$$b_n b_{n-1} \dots b_1 b_0 . b_{-1} \dots b_{-m} = \sum_{i=-m}^n b_i 2^i.$$

Például $110.101_2 = 6.625$

Konverzió 10-es számrendszerből 2-es számrendszerbe

- egészek esetén ismételt 2-vel való maradékos osztás,
- törtrész esetén ismételt 2-vel való szorzás.

Például 106 bináris alakja:

$$106 = 2 \cdot 53 + 0 \rightarrow 0$$

$$53 = 2 \cdot 26 + 1 \rightarrow 1$$

$$26 = 2 \cdot 13 + 0 \rightarrow 0$$

$$13 = 2 \cdot 6 + 1 \rightarrow 1$$

$$6 = 2 \cdot 3 + 0$$

Konverzió 2-es számrendszerből 10-es számrendszerbe:

$$b_n b_{n-1} \dots b_1 b_0 . b_{-1} \dots b_{-m} = \sum_{i=-m}^n b_i 2^i.$$

Például $110.101_2 = 6.625$

Konverzió 10-es számrendszerből 2-es számrendszerbe

- egészek esetén ismételt 2-vel való maradékos osztás,
- törtrész esetén ismételt 2-vel való szorzás.

Például 106 bináris alakja:

$$106 = 2 \cdot 53 + 0 \rightarrow 0$$

$$53 = 2 \cdot 26 + 1 \rightarrow 1$$

$$26 = 2 \cdot 13 + 0 \rightarrow 0$$

$$13 = 2 \cdot 6 + 1 \rightarrow 1$$

$$6 = 2 \cdot 3 + 0 \rightarrow 0$$

Konverzió 2-es számrendszerből 10-es számrendszerbe:

$$b_n b_{n-1} \dots b_1 b_0 . b_{-1} \dots b_{-m} = \sum_{i=-m}^n b_i 2^i.$$

Például $110.101_2 = 6.625$

Konverzió 10-es számrendszerből 2-es számrendszerbe

- egészek esetén ismételt 2-vel való maradékos osztás,
- törtrész esetén ismételt 2-vel való szorzás.

Például 106 bináris alakja:

$$106 = 2 \cdot 53 + 0 \rightarrow 0$$

$$53 = 2 \cdot 26 + 1 \rightarrow 1$$

$$26 = 2 \cdot 13 + 0 \rightarrow 0$$

$$13 = 2 \cdot 6 + 1 \rightarrow 1$$

$$6 = 2 \cdot 3 + 0 \rightarrow 0$$

$$3 = 2 \cdot 1 + 1$$

Konverzió 2-es számrendszerből 10-es számrendszerbe:

$$b_n b_{n-1} \dots b_1 b_0 . b_{-1} \dots b_{-m} = \sum_{i=-m}^n b_i 2^i.$$

Például $110.101_2 = 6.625$

Konverzió 10-es számrendszerből 2-es számrendszerbe

- egészek esetén ismételt 2-vel való maradékos osztás,
- törtrész esetén ismételt 2-vel való szorzás.

Például 106 bináris alakja:

$$106 = 2 \cdot 53 + 0 \rightarrow 0$$

$$53 = 2 \cdot 26 + 1 \rightarrow 1$$

$$26 = 2 \cdot 13 + 0 \rightarrow 0$$

$$13 = 2 \cdot 6 + 1 \rightarrow 1$$

$$6 = 2 \cdot 3 + 0 \rightarrow 0$$

$$3 = 2 \cdot 1 + 1 \rightarrow 1$$

Konverzió 2-es számrendszerből 10-es számrendszerbe:

$$b_n b_{n-1} \dots b_1 b_0 . b_{-1} \dots b_{-m} = \sum_{i=-m}^n b_i 2^i.$$

Például $110.101_2 = 6.625$

Konverzió 10-es számrendszerből 2-es számrendszerbe

- egészek esetén ismételt 2-vel való maradékos osztás,
- törtrész esetén ismételt 2-vel való szorzás.

Például 106 bináris alakja:

$$106 = 2 \cdot 53 + 0 \rightarrow 0$$

$$53 = 2 \cdot 26 + 1 \rightarrow 1$$

$$26 = 2 \cdot 13 + 0 \rightarrow 0$$

$$13 = 2 \cdot 6 + 1 \rightarrow 1$$

$$6 = 2 \cdot 3 + 0 \rightarrow 0$$

$$3 = 2 \cdot 1 + 1 \rightarrow 1$$

$$1 = 2 \cdot 0 + 1$$

Konverzió 2-es számrendszerből 10-es számrendszerbe:

$$b_n b_{n-1} \dots b_1 b_0 . b_{-1} \dots b_{-m} = \sum_{i=-m}^n b_i 2^i.$$

Például $110.101_2 = 6.625$

Konverzió 10-es számrendszerből 2-es számrendszerbe

- egészek esetén ismételt 2-vel való maradékos osztás,
- törtrész esetén ismételt 2-vel való szorzás.

Például 106 bináris alakja:

$$106 = 2 \cdot 53 + 0 \rightarrow 0$$

$$53 = 2 \cdot 26 + 1 \rightarrow 1$$

$$26 = 2 \cdot 13 + 0 \rightarrow 0$$

$$13 = 2 \cdot 6 + 1 \rightarrow 1$$

$$6 = 2 \cdot 3 + 0 \rightarrow 0$$

$$3 = 2 \cdot 1 + 1 \rightarrow 1$$

$$1 = 2 \cdot 0 + 1 \rightarrow 1$$

Konverzió 2-es számrendszerből 10-es számrendszerbe:

$$b_n b_{n-1} \dots b_1 b_0 . b_{-1} \dots b_{-m} = \sum_{i=-m}^n b_i 2^i.$$

Például $110.101_2 = 6.625$

Konverzió 10-es számrendszerből 2-es számrendszerbe

- egészek esetén ismételt 2-vel való maradékos osztás,
- törtrész esetén ismételt 2-vel való szorzás.

Például 106 bináris alakja:

$$106 = 2 \cdot 53 + 0 \rightarrow 0$$

$$53 = 2 \cdot 26 + 1 \rightarrow 1$$

$$26 = 2 \cdot 13 + 0 \rightarrow 0$$

$$13 = 2 \cdot 6 + 1 \rightarrow 1$$

$$6 = 2 \cdot 3 + 0 \rightarrow 0$$

$$3 = 2 \cdot 1 + 1 \rightarrow 1$$

$$1 = 2 \cdot 0 + 1 \rightarrow 1$$

tehát a bináris alak 1101010_2 .

Konverzió 2-es számrendszerből 10-es számrendszerbe:

$$b_n b_{n-1} \dots b_1 b_0 . b_{-1} \dots b_{-m} = \sum_{i=-m}^n b_i 2^i.$$

Például $110.101_2 = 6.625$

Konverzió 10-es számrendszerből 2-es számrendszerbe

- egészek esetén ismételt 2-vel való maradékos osztás,
- törtrész esetén ismételt 2-vel való szorzás.

Például 106 bináris alakja:

$$106 = 2 \cdot 53 + 0 \rightarrow 0$$

$$53 = 2 \cdot 26 + 1 \rightarrow 1$$

$$26 = 2 \cdot 13 + 0 \rightarrow 0$$

$$13 = 2 \cdot 6 + 1 \rightarrow 1$$

$$6 = 2 \cdot 3 + 0 \rightarrow 0$$

$$3 = 2 \cdot 1 + 1 \rightarrow 1$$

$$1 = 2 \cdot 0 + 1 \rightarrow 1$$

tehát a bináris alak 1101010_2 .

$$\begin{array}{r|l} 106 & 2 \\ \hline 53 & 0 \end{array}$$

Konverzió 2-es számrendszerből 10-es számrendszerbe:

$$b_n b_{n-1} \dots b_1 b_0 . b_{-1} \dots b_{-m} = \sum_{i=-m}^n b_i 2^i.$$

Például $110.101_2 = 6.625$

Konverzió 10-es számrendszerből 2-es számrendszerbe

- egészek esetén ismételt 2-vel való maradékos osztás,
- törtrész esetén ismételt 2-vel való szorzás.

Például 106 bináris alakja:

$$106 = 2 \cdot 53 + 0 \rightarrow 0$$

$$53 = 2 \cdot 26 + 1 \rightarrow 1$$

$$26 = 2 \cdot 13 + 0 \rightarrow 0$$

$$13 = 2 \cdot 6 + 1 \rightarrow 1$$

$$6 = 2 \cdot 3 + 0 \rightarrow 0$$

$$3 = 2 \cdot 1 + 1 \rightarrow 1$$

$$1 = 2 \cdot 0 + 1 \rightarrow 1$$

tehát a bináris alak 1101010_2 .

106	2
53	0
26	1

Konverzió 2-es számrendszerből 10-es számrendszerbe:

$$b_n b_{n-1} \dots b_1 b_0 . b_{-1} \dots b_{-m} = \sum_{i=-m}^n b_i 2^i.$$

Például $110.101_2 = 6.625$

Konverzió 10-es számrendszerből 2-es számrendszerbe

- egészek esetén ismételt 2-vel való maradékos osztás,
- törtrész esetén ismételt 2-vel való szorzás.

Például 106 bináris alakja:

$$106 = 2 \cdot 53 + 0 \rightarrow 0$$

$$53 = 2 \cdot 26 + 1 \rightarrow 1$$

$$26 = 2 \cdot 13 + 0 \rightarrow 0$$

$$13 = 2 \cdot 6 + 1 \rightarrow 1$$

$$6 = 2 \cdot 3 + 0 \rightarrow 0$$

$$3 = 2 \cdot 1 + 1 \rightarrow 1$$

$$1 = 2 \cdot 0 + 1 \rightarrow 1$$

106	2
53	0
26	1
13	0

tehát a bináris alak 1101010_2 .

Konverzió 2-es számrendszerből 10-es számrendszerbe:

$$b_n b_{n-1} \dots b_1 b_0 . b_{-1} \dots b_{-m} = \sum_{i=-m}^n b_i 2^i.$$

Például $110.101_2 = 6.625$

Konverzió 10-es számrendszerből 2-es számrendszerbe

- egészek esetén ismételt 2-vel való maradékos osztás,
- törtrész esetén ismételt 2-vel való szorzás.

Például 106 bináris alakja:

$$106 = 2 \cdot 53 + 0 \rightarrow 0$$

$$53 = 2 \cdot 26 + 1 \rightarrow 1$$

$$26 = 2 \cdot 13 + 0 \rightarrow 0$$

$$13 = 2 \cdot 6 + 1 \rightarrow 1$$

$$6 = 2 \cdot 3 + 0 \rightarrow 0$$

$$3 = 2 \cdot 1 + 1 \rightarrow 1$$

$$1 = 2 \cdot 0 + 1 \rightarrow 1$$

106	2
53	0
26	1
13	0
6	1

tehát a bináris alak 1101010 .

Konverzió 2-es számrendszerből 10-es számrendszerbe:

$$b_n b_{n-1} \dots b_1 b_0 . b_{-1} \dots b_{-m} = \sum_{i=-m}^n b_i 2^i.$$

Például $110.101_2 = 6.625$

Konverzió 10-es számrendszerből 2-es számrendszerbe

- egészek esetén ismételt 2-vel való maradékos osztás,
- törtrész esetén ismételt 2-vel való szorzás.

Például 106 bináris alakja:

$$106 = 2 \cdot 53 + 0 \rightarrow 0$$

$$53 = 2 \cdot 26 + 1 \rightarrow 1$$

$$26 = 2 \cdot 13 + 0 \rightarrow 0$$

$$13 = 2 \cdot 6 + 1 \rightarrow 1$$

$$6 = 2 \cdot 3 + 0 \rightarrow 0$$

$$3 = 2 \cdot 1 + 1 \rightarrow 1$$

$$1 = 2 \cdot 0 + 1 \rightarrow 1$$

106	2
53	0
26	1
13	0
6	1
3	0

tehát a bináris alak 1101010_2 .

Konverzió 2-es számrendszerből 10-es számrendszerbe:

$$b_n b_{n-1} \dots b_1 b_0 . b_{-1} \dots b_{-m} = \sum_{i=-m}^n b_i 2^i.$$

Például $110.101_2 = 6.625$

Konverzió 10-es számrendszerből 2-es számrendszerbe

- egészek esetén ismételt 2-vel való maradékos osztás,
- törtrész esetén ismételt 2-vel való szorzás.

Például 106 bináris alakja:

$$106 = 2 \cdot 53 + 0 \rightarrow 0$$

$$53 = 2 \cdot 26 + 1 \rightarrow 1$$

$$26 = 2 \cdot 13 + 0 \rightarrow 0$$

$$13 = 2 \cdot 6 + 1 \rightarrow 1$$

$$6 = 2 \cdot 3 + 0 \rightarrow 0$$

$$3 = 2 \cdot 1 + 1 \rightarrow 1$$

$$1 = 2 \cdot 0 + 1 \rightarrow 1$$

106	2
53	0
26	1
13	0
6	1
3	0
1	1

tehát a bináris alak 1101010_2 .

Konverzió 2-es számrendszerből 10-es számrendszerbe:

$$b_n b_{n-1} \dots b_1 b_0 . b_{-1} \dots b_{-m} = \sum_{i=-m}^n b_i 2^i.$$

Például $110.101_2 = 6.625$

Konverzió 10-es számrendszerből 2-es számrendszerbe

- egészek esetén ismételt 2-vel való maradékos osztás,
- törtrész esetén ismételt 2-vel való szorzás.

Például 106 bináris alakja:

$$106 = 2 \cdot 53 + 0 \rightarrow 0$$

$$53 = 2 \cdot 26 + 1 \rightarrow 1$$

$$26 = 2 \cdot 13 + 0 \rightarrow 0$$

$$13 = 2 \cdot 6 + 1 \rightarrow 1$$

$$6 = 2 \cdot 3 + 0 \rightarrow 0$$

$$3 = 2 \cdot 1 + 1 \rightarrow 1$$

$$1 = 2 \cdot 0 + 1 \rightarrow 1$$

tehát a bináris alak 1101010_2 .

106	2
53	0
26	1
13	0
6	1
3	0
1	1
0	1

Konverzió 2-es számrendszerből 10-es számrendszerbe:

$$b_n b_{n-1} \dots b_1 b_0 . b_{-1} \dots b_{-m} = \sum_{i=-m}^n b_i 2^i.$$

Például $110.101_2 = 6.625$

Konverzió 10-es számrendszerből 2-es számrendszerbe

- egészek esetén ismételt 2-vel való maradékos osztás,
- törtrész esetén ismételt 2-vel való szorzás.

Például 106 bináris alakja:

$$106 = 2 \cdot 53 + 0 \rightarrow 0$$

$$53 = 2 \cdot 26 + 1 \rightarrow 1$$

$$26 = 2 \cdot 13 + 0 \rightarrow 0$$

$$13 = 2 \cdot 6 + 1 \rightarrow 1$$

$$6 = 2 \cdot 3 + 0 \rightarrow 0$$

$$3 = 2 \cdot 1 + 1 \rightarrow 1$$

$$1 = 2 \cdot 0 + 1 \rightarrow 1$$

tehát a bináris alak 1101010_2 .

106	2
53	0
26	1
13	0
6	1
3	0
1	1
0	1

Példa

Hogyan konvertálunk tizedes törtet binárisra?

Példa

Hogyan konvertálunk tizedes törtet binárisra? Pl. írjuk fel 0.3 bináris alakjának tizedespont utáni első 6 jegyét!

Példa

Hogyan konvertálunk tizedes törtet binárisra? Pl. írjuk fel 0.3 bináris alakjának tizedespont utáni első 6 jegyét!

Megoldás: A tizedespont utáni jegyek jelentése bináris esetben $1/2$, $1/4, \dots, 1/2^n, \dots$. Pl. a bináris 0.1011001 számot mindig 2-vel szorozva az eredmény egész része rendre 1, 0, 1, 1, 0, 0, 1.

Példa

Hogyan konvertálunk tizedes törtet binárisra? Pl. írjuk fel 0.3 bináris alakjának tizedespont utáni első 6 jegyét!

Megoldás: A tizedespont utáni jegyek jelentése bináris esetben $1/2$, $1/4, \dots, 1/2^n, \dots$. Pl. a bináris 0.1011001 számot mindig 2-vel szorozva az eredmény egész része rendre 1, 0, 1, 1, 0, 0, 1. Ezt használva:
 $0.3 \cdot 2 = 0.6$

Példa

Hogyan konvertálunk tizedes törtet binárisra? Pl. írjuk fel 0.3 bináris alakjának tizedespont utáni első 6 jegyét!

Megoldás: A tizedespont utáni jegyek jelentése bináris esetben $1/2$, $1/4, \dots, 1/2^n, \dots$. Pl. a bináris 0.1011001 számot mindig 2-vel szorozva az eredmény egész része rendre 1, 0, 1, 1, 0, 0, 1. Ezt használva:
 $0.3 \cdot 2 = 0.6 \rightarrow 0$

Példa

Hogyan konvertálunk tizedes törtet binárisra? Pl. írjuk fel 0.3 bináris alakjának tizedespont utáni első 6 jegyét!

Megoldás: A tizedespont utáni jegyek jelentése bináris esetben $1/2$, $1/4, \dots, 1/2^n, \dots$. Pl. a bináris 0.1011001 számot mindig 2-vel szorozva az eredmény egész része rendre 1, 0, 1, 1, 0, 0, 1. Ezt használva:

$$0.3 \cdot 2 = 0.6 \rightarrow 0$$

$$0.6 \cdot 2 = 1.2$$

Példa

Hogyan konvertálunk tizedes törtet binárisra? Pl. írjuk fel 0.3 bináris alakjának tizedespont utáni első 6 jegyét!

Megoldás: A tizedespont utáni jegyek jelentése bináris esetben $1/2$, $1/4, \dots, 1/2^n, \dots$. Pl. a bináris 0.1011001 számot mindig 2-vel szorozva az eredmény egész része rendre 1, 0, 1, 1, 0, 0, 1. Ezt használva:

$$0.3 \cdot 2 = 0.6 \rightarrow 0$$

$$0.6 \cdot 2 = 1.2 \rightarrow 1$$

Példa

Hogyan konvertálunk tizedes törtet binárisra? Pl. írjuk fel 0.3 bináris alakjának tizedespont utáni első 6 jegyét!

Megoldás: A tizedespont utáni jegyek jelentése bináris esetben $1/2$, $1/4, \dots, 1/2^n, \dots$. Pl. a bináris 0.1011001 számot mindig 2-vel szorozva az eredmény egész része rendre 1, 0, 1, 1, 0, 0, 1. Ezt használva:

$$0.3 \cdot 2 = 0.6 \rightarrow 0$$

$$0.6 \cdot 2 = 1.2 \rightarrow 1$$

$$0.2 \cdot 2 = 0.4$$

Példa

Hogyan konvertálunk tizedes törtet binárisra? Pl. írjuk fel 0.3 bináris alakjának tizedespont utáni első 6 jegyét!

Megoldás: A tizedespont utáni jegyek jelentése bináris esetben $1/2$, $1/4, \dots, 1/2^n, \dots$. Pl. a bináris 0.1011001 számot mindig 2-vel szorozva az eredmény egész része rendre 1, 0, 1, 1, 0, 0, 1. Ezt használva:

$$0.3 \cdot 2 = 0.6 \rightarrow 0$$

$$0.6 \cdot 2 = 1.2 \rightarrow 1$$

$$0.2 \cdot 2 = 0.4 \rightarrow 0$$

Példa

Hogyan konvertálunk tizedes törtet binárisra? Pl. írjuk fel 0.3 bináris alakjának tizedespont utáni első 6 jegyét!

Megoldás: A tizedespont utáni jegyek jelentése bináris esetben $1/2$, $1/4, \dots, 1/2^n, \dots$. Pl. a bináris 0.1011001 számot mindig 2-vel szorozva az eredmény egész része rendre 1, 0, 1, 1, 0, 0, 1. Ezt használva:

$$0.3 \cdot 2 = 0.6 \rightarrow 0$$

$$0.6 \cdot 2 = 1.2 \rightarrow 1$$

$$0.2 \cdot 2 = 0.4 \rightarrow 0$$

$$0.4 \cdot 2 = 0.8$$

Példa

Hogyan konvertálunk tizedes törtet binárisra? Pl. írjuk fel 0.3 bináris alakjának tizedespont utáni első 6 jegyét!

Megoldás: A tizedespont utáni jegyek jelentése bináris esetben $1/2$, $1/4, \dots, 1/2^n, \dots$. Pl. a bináris 0.1011001 számot mindig 2-vel szorozva az eredmény egész része rendre 1, 0, 1, 1, 0, 0, 1. Ezt használva:

$$0.3 \cdot 2 = 0.6 \rightarrow 0$$

$$0.6 \cdot 2 = 1.2 \rightarrow 1$$

$$0.2 \cdot 2 = 0.4 \rightarrow 0$$

$$0.4 \cdot 2 = 0.8 \rightarrow 0$$

Példa

Hogyan konvertálunk tizedes törtet binárisra? Pl. írjuk fel 0.3 bináris alakjának tizedespont utáni első 6 jegyét!

Megoldás: A tizedespont utáni jegyek jelentése bináris esetben $1/2$, $1/4, \dots, 1/2^n, \dots$. Pl. a bináris 0.1011001 számot mindig 2-vel szorozva az eredmény egész része rendre 1, 0, 1, 1, 0, 0, 1. Ezt használva:

$$0.3 \cdot 2 = 0.6 \rightarrow 0$$

$$0.6 \cdot 2 = 1.2 \rightarrow 1$$

$$0.2 \cdot 2 = 0.4 \rightarrow 0$$

$$0.4 \cdot 2 = 0.8 \rightarrow 0$$

$$0.8 \cdot 2 = 1.6$$

Példa

Hogyan konvertálunk tizedes törtet binárisra? Pl. írjuk fel 0.3 bináris alakjának tizedespont utáni első 6 jegyét!

Megoldás: A tizedespont utáni jegyek jelentése bináris esetben $1/2$, $1/4$, ..., $1/2^n$, Pl. a bináris 0.1011001 számot mindig 2-vel szorozva az eredmény egész része rendre 1, 0, 1, 1, 0, 0, 1. Ezt használva:

$$0.3 \cdot 2 = 0.6 \rightarrow 0$$

$$0.6 \cdot 2 = 1.2 \rightarrow 1$$

$$0.2 \cdot 2 = 0.4 \rightarrow 0$$

$$0.4 \cdot 2 = 0.8 \rightarrow 0$$

$$0.8 \cdot 2 = 1.6 \rightarrow 1$$

Példa

Hogyan konvertálunk tizedes törtet binárisra? Pl. írjuk fel 0.3 bináris alakjának tizedespont utáni első 6 jegyét!

Megoldás: A tizedespont utáni jegyek jelentése bináris esetben $1/2$, $1/4, \dots, 1/2^n, \dots$. Pl. a bináris 0.1011001 számot mindig 2-vel szorozva az eredmény egész része rendre 1, 0, 1, 1, 0, 0, 1. Ezt használva:

$$0.3 \cdot 2 = 0.6 \rightarrow 0$$

$$0.6 \cdot 2 = 1.2 \rightarrow 1$$

$$0.2 \cdot 2 = 0.4 \rightarrow 0$$

$$0.4 \cdot 2 = 0.8 \rightarrow 0$$

$$0.8 \cdot 2 = 1.6 \rightarrow 1$$

$$0.6 \cdot 2 = 1.2$$

Példa

Hogyan konvertálunk tizedes törtet binárisra? Pl. írjuk fel 0.3 bináris alakjának tizedespont utáni első 6 jegyét!

Megoldás: A tizedespont utáni jegyek jelentése bináris esetben $1/2$, $1/4, \dots, 1/2^n, \dots$. Pl. a bináris 0.1011001 számot mindig 2-vel szorozva az eredmény egész része rendre 1, 0, 1, 1, 0, 0, 1. Ezt használva:

$$0.3 \cdot 2 = 0.6 \rightarrow 0$$

$$0.6 \cdot 2 = 1.2 \rightarrow 1$$

$$0.2 \cdot 2 = 0.4 \rightarrow 0$$

$$0.4 \cdot 2 = 0.8 \rightarrow 0$$

$$0.8 \cdot 2 = 1.6 \rightarrow 1$$

$$0.6 \cdot 2 = 1.2 \rightarrow 1$$

Példa

Hogyan konvertálunk tizedes törtet binárisra? Pl. írjuk fel 0.3 bináris alakjának tizedespont utáni első 6 jegyét!

Megoldás: A tizedespont utáni jegyek jelentése bináris esetben $1/2$, $1/4, \dots, 1/2^n, \dots$. Pl. a bináris 0.1011001 számot mindig 2-vel szorozva az eredmény egész része rendre 1, 0, 1, 1, 0, 0, 1. Ezt használva:

$$0.3 \cdot 2 = 0.6 \rightarrow 0$$

$$0.6 \cdot 2 = 1.2 \rightarrow 1$$

$$0.2 \cdot 2 = 0.4 \rightarrow 0$$

$$0.4 \cdot 2 = 0.8 \rightarrow 0$$

$$0.8 \cdot 2 = 1.6 \rightarrow 1$$

$$0.6 \cdot 2 = 1.2 \rightarrow 1$$

Azaz 0.3 bináris alakja 0.010011, sőt az is látszik, hogy a végtelen bináris alak: $0.0\dot{1}00\dot{1}$.

Példa

Hogyan konvertálunk tizedes törtet binárisra? Pl. írjuk fel 0.3 bináris alakjának tizedespont utáni első 6 jegyét!

Megoldás: A tizedespont utáni jegyek jelentése bináris esetben $1/2$, $1/4, \dots, 1/2^n, \dots$. Pl. a bináris 0.1011001 számot mindig 2-vel szorozva az eredmény egész része rendre 1, 0, 1, 1, 0, 0, 1. Ezt használva:

$$0.3 \cdot 2 = 0.6 \rightarrow 0$$

$$0.6 \cdot 2 = 1.2 \rightarrow 1$$

$$0.2 \cdot 2 = 0.4 \rightarrow 0$$

$$0.4 \cdot 2 = 0.8 \rightarrow 0$$

$$0.8 \cdot 2 = 1.6 \rightarrow 1$$

$$0.6 \cdot 2 = 1.2 \rightarrow 1$$

Azaz 0.3 bináris alakja 0.010011, sőt az is látszik, hogy a végtelen bináris alak: $0.0\dot{1}00\dot{1}$.

0.3		2
0.6		0

Példa

Hogyan konvertálunk tizedes törtet binárisra? Pl. írjuk fel 0.3 bináris alakjának tizedespont utáni első 6 jegyét!

Megoldás: A tizedespont utáni jegyek jelentése bináris esetben $1/2$, $1/4, \dots, 1/2^n, \dots$. Pl. a bináris 0.1011001 számot mindig 2-vel szorozva az eredmény egész része rendre 1, 0, 1, 1, 0, 0, 1. Ezt használva:

$$0.3 \cdot 2 = 0.6 \rightarrow 0$$

$$0.6 \cdot 2 = 1.2 \rightarrow 1$$

$$0.2 \cdot 2 = 0.4 \rightarrow 0$$

$$0.4 \cdot 2 = 0.8 \rightarrow 0$$

$$0.8 \cdot 2 = 1.6 \rightarrow 1$$

$$0.6 \cdot 2 = 1.2 \rightarrow 1$$

Azaz 0.3 bináris alakja 0.010011,
sőt az is látszik, hogy a végtelen
bináris alak: $0.0\bar{1}00\bar{1}$.

0.3	2
0.6	0
1.2	1

Példa

Hogyan konvertálunk tizedes törtet binárisra? Pl. írjuk fel 0.3 bináris alakjának tizedespont utáni első 6 jegyét!

Megoldás: A tizedespont utáni jegyek jelentése bináris esetben $1/2$, $1/4, \dots, 1/2^n, \dots$. Pl. a bináris 0.1011001 számot mindig 2-vel szorozva az eredmény egész része rendre 1, 0, 1, 1, 0, 0, 1. Ezt használva:

$$0.3 \cdot 2 = 0.6 \rightarrow 0$$

$$0.6 \cdot 2 = 1.2 \rightarrow 1$$

$$0.2 \cdot 2 = 0.4 \rightarrow 0$$

$$0.4 \cdot 2 = 0.8 \rightarrow 0$$

$$0.8 \cdot 2 = 1.6 \rightarrow 1$$

$$0.6 \cdot 2 = 1.2 \rightarrow 1$$

Azaz 0.3 bináris alakja 0.010011,
sőt az is látszik, hogy a végtelen
bináris alak: $0.0\bar{1}00\bar{1}$.

0.3	2
0.6	0
1.2	1
0.4	0

Példa

Hogyan konvertálunk tizedes törtet binárisra? Pl. írjuk fel 0.3 bináris alakjának tizedespont utáni első 6 jegyét!

Megoldás: A tizedespont utáni jegyek jelentése bináris esetben $1/2$, $1/4, \dots, 1/2^n, \dots$. Pl. a bináris 0.1011001 számot mindig 2-vel szorozva az eredmény egész része rendre 1, 0, 1, 1, 0, 0, 1. Ezt használva:

$$0.3 \cdot 2 = 0.6 \rightarrow 0$$

$$0.6 \cdot 2 = 1.2 \rightarrow 1$$

$$0.2 \cdot 2 = 0.4 \rightarrow 0$$

$$0.4 \cdot 2 = 0.8 \rightarrow 0$$

$$0.8 \cdot 2 = 1.6 \rightarrow 1$$

$$0.6 \cdot 2 = 1.2 \rightarrow 1$$

0.3	2
0.6	0
1.2	1
0.4	0
0.8	0

Azaz 0.3 bináris alakja 0.010011,
sőt az is látszik, hogy a végtelen
bináris alak: $0.0\bar{1}00\bar{1}$.

Példa

Hogyan konvertálunk tizedes törtet binárisra? Pl. írjuk fel 0.3 bináris alakjának tizedespont utáni első 6 jegyét!

Megoldás: A tizedespont utáni jegyek jelentése bináris esetben $1/2$, $1/4, \dots, 1/2^n, \dots$. Pl. a bináris 0.1011001 számot mindig 2-vel szorozva az eredmény egész része rendre 1, 0, 1, 1, 0, 0, 1. Ezt használva:

$$0.3 \cdot 2 = 0.6 \rightarrow 0$$

$$0.6 \cdot 2 = 1.2 \rightarrow 1$$

$$0.2 \cdot 2 = 0.4 \rightarrow 0$$

$$0.4 \cdot 2 = 0.8 \rightarrow 0$$

$$0.8 \cdot 2 = 1.6 \rightarrow 1$$

$$0.6 \cdot 2 = 1.2 \rightarrow 1$$

Azaz 0.3 bináris alakja 0.010011,
sőt az is látszik, hogy a végtelen
bináris alak: $0.0\bar{1}00\bar{1}$.

0.3	2
0.6	0
1.2	1
0.4	0
0.8	0
1.6	1

Példa

Hogyan konvertálunk tizedes törtet binárisra? Pl. írjuk fel 0.3 bináris alakjának tizedespont utáni első 6 jegyét!

Megoldás: A tizedespont utáni jegyek jelentése bináris esetben $1/2$, $1/4, \dots, 1/2^n, \dots$. Pl. a bináris 0.1011001 számot mindig 2-vel szorozva az eredmény egész része rendre 1, 0, 1, 1, 0, 0, 1. Ezt használva:

$$0.3 \cdot 2 = 0.6 \rightarrow 0$$

$$0.6 \cdot 2 = 1.2 \rightarrow 1$$

$$0.2 \cdot 2 = 0.4 \rightarrow 0$$

$$0.4 \cdot 2 = 0.8 \rightarrow 0$$

$$0.8 \cdot 2 = 1.6 \rightarrow 1$$

$$0.6 \cdot 2 = 1.2 \rightarrow 1$$

Azaz 0.3 bináris alakja 0.010011,
sőt az is látszik, hogy a végtelen
bináris alak: $0.0\bar{1}00\bar{1}$.

0.3	2
0.6	0
1.2	1
0.4	0
0.8	0
1.6	1
1.2	1

Példa

Hogyan konvertálunk tizedes törtet binárisra? Pl. írjuk fel 0.3 bináris alakjának tizedespont utáni első 6 jegyét!

Megoldás: A tizedespont utáni jegyek jelentése bináris esetben $1/2$, $1/4, \dots, 1/2^n, \dots$. Pl. a bináris 0.1011001 számot mindig 2-vel szorozva az eredmény egész része rendre 1, 0, 1, 1, 0, 0, 1. Ezt használva:

$$0.3 \cdot 2 = 0.6 \rightarrow 0$$

$$0.6 \cdot 2 = 1.2 \rightarrow 1$$

$$0.2 \cdot 2 = 0.4 \rightarrow 0$$

$$0.4 \cdot 2 = 0.8 \rightarrow 0$$

$$0.8 \cdot 2 = 1.6 \rightarrow 1$$

$$0.6 \cdot 2 = 1.2 \rightarrow 1$$

Azaz 0.3 bináris alakja 0.010011,
sőt az is látszik, hogy a végtelen
bináris alak: $0.0\bar{1}00\bar{1}$.

0.3	2
0.6	0
1.2	1
0.4	0
0.8	0
1.6	1
1.2	1

Hexadecimális számrendszer

Hexadecimális (16-os számrendszerbeli) számok:

bin	hex	bin	hex
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

Hexadecimális számrendszer

Hexadecimális (16-os számrendszerbeli) számok:

bin	hex	bin	hex
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

Például $0011\ 1100\ 1111\ 1010 = 0x3CFA$.

Definíció (bit)

Egy bit az az információ, amit egy (ismeretlen) eldönthető kérdés megválaszolása nyújt.

Definíció (bit)

Egy bit az az információ, amit egy (ismeretlen) eldönthető kérdés megválaszolása nyújt.

Definíció (byte)

A számítógép memóriájának egyetlen cellájának mérete. Ennél kisebb adatot nem lehet megcímezni a memóriában.

Definíció (bit)

Egy bit az az információ, amit egy (ismeretlen) eldönthető kérdés megválaszolása nyújt.

Definíció (byte)

A számítógép memóriájának egyetlen cellájának mérete. Ennél kisebb adatot nem lehet megcímezni a memóriában.

Általában 1 byte 8 bit (256 különböző érték).

Definíció (bit)

Egy bit az az információ, amit egy (ismeretlen) eldönthető kérdés megválaszolása nyújt.

Definíció (byte)

A számítógép memóriájának egyetlen cellájának mérete. Ennél kisebb adatot nem lehet megcímezni a memóriában.

Általában 1 byte 8 bit (256 különböző érték).

Definíció (szó (word))

Egy tipikus utasítás egyetlen változójának mérete. A memória mutatók nagysága, általában a regiszterek mérete.

Definíció (bit)

Egy bit az az információ, amit egy (ismeretlen) eldönthető kérdés megválaszolása nyújt.

Definíció (byte)

A számítógép memóriájának egyetlen cellájának mérete. Ennél kisebb adatot nem lehet megcímezni a memóriában.

Általában 1 byte 8 bit (256 különböző érték).

Definíció (szó (word))

Egy tipikus utasítás egyetlen változójának mérete. A memória mutatók nagysága, általában a regiszterek mérete.

Tipikusan 64 bit, azaz 8 byte.

Egyes komplement számábrázolás

n-biten: az első bit az előjel.

Egyes komplement számábrázolás

n -biten: az első bit az előjel. Az ábrázolható számok tartománya:
 $-2^{n-1} + 1$ -től $2^{n-1} - 1$ -ig.

Egyes komplement számábrázolás

***n*-biten**: az első bit az előjel. Az ábrázolható számok tartománya:
 $-2^{n-1} + 1$ -től $2^{n-1} - 1$ -ig.

Például 4 biten: -7 -től 7 -ig ábrázolhatók a számok.

1001 $\rightarrow -1$

1100 $\rightarrow -4$

1111 $\rightarrow -7$

1000 $\rightarrow -0$

0000 $\rightarrow +0$

Egyes komplementens számábrázolás

***n*-biten:** az első bit az előjel. Az ábrázolható számok tartománya:
 $-2^{n-1} + 1$ -től $2^{n-1} - 1$ -ig.

Például 4 biten: -7 -től 7 -ig ábrázolhatók a számok.

1001 $\rightarrow -1$

1100 $\rightarrow -4$

1111 $\rightarrow -7$

1000 $\rightarrow -0$

0000 $\rightarrow +0$

Hátránya: van $+0$ és -0 .

Kettes komplementens számábrázolás

***n*-biten**: előjelbites számábrázolást keresünk, ahol nincs $+0$ és -0 .

$$\bar{x} = \begin{cases} x & \text{ha } x \text{ nem negatív,} \\ 2^n - |x| & \text{ha } x \text{ negatív.} \end{cases}$$

***n*-biten**: előjelbites számábrázolást keresünk, ahol nincs $+0$ és -0 .

$$\bar{x} = \begin{cases} x & \text{ha } x \text{ nem negatív,} \\ 2^n - |x| & \text{ha } x \text{ negatív.} \end{cases}$$

$2^n - |x|$ kiszámítása n -bites szavak közti bitműveletekkel: $|x|$ bitenkénti komplemente $+ 1$, ugyanis

$$2^n - |x| = (2^n - 1) - |x| + 1 = 11 \dots 1_2 - |x| + 1.$$

Kettes komplementes számábrázolás

n -biten: előjelbites számábrázolást keresünk, ahol nincs $+0$ és -0 .

$$\bar{x} = \begin{cases} x & \text{ha } x \text{ nem negatív,} \\ 2^n - |x| & \text{ha } x \text{ negatív.} \end{cases}$$

$2^n - |x|$ kiszámítása n -bites szavak közti bitműveletekkel: $|x|$ bitenkénti komplementese $+1$, ugyanis

$2^n - |x| = (2^n - 1) - |x| + 1 = 11 \dots 1_2 - |x| + 1$. Mivel $|x| = 2^n - (2^n - |x|)$, ezért x értékének meghatározása $\bar{x}$ -ből ugyanígy történik, azaz ha az első bit egyes, $|x|$ értéke $= \bar{x}$ komplementese $+1$.

Kettes komplementum számábrázolás

n -biten: előjelbites számábrázolást keresünk, ahol nincs $+0$ és -0 .

$$\bar{x} = \begin{cases} x & \text{ha } x \text{ nem negatív,} \\ 2^n - |x| & \text{ha } x \text{ negatív.} \end{cases}$$

$2^n - |x|$ kiszámítása n -bites szavak közti bitműveletekkel: $|x|$ bitenkénti komplementum + 1, ugyanis

$2^n - |x| = (2^n - 1) - |x| + 1 = 11 \dots 1_2 - |x| + 1$. Mivel $|x| = 2^n - (2^n - |x|)$, ezért x értékének meghatározása $\bar{x}$ -ből ugyanígy történik, azaz ha az első bit egyes, $|x|$ értéke $= \bar{x}$ komplementum + 1.

-1 alakja

Kettes komplementens számábrázolás

n -biten: előjelbites számábrázolást keresünk, ahol nincs $+0$ és -0 .

$$\bar{x} = \begin{cases} x & \text{ha } x \text{ nem negatív,} \\ 2^n - |x| & \text{ha } x \text{ negatív.} \end{cases}$$

$2^n - |x|$ kiszámítása n -bites szavak közti bitműveletekkel: $|x|$ bitenkénti komplementense $+1$, ugyanis

$2^n - |x| = (2^n - 1) - |x| + 1 = 11 \dots 1_2 - |x| + 1$. Mivel $|x| = 2^n - (2^n - |x|)$, ezért x értékének meghatározása $\bar{x}$ -ből ugyanígy történik, azaz ha az első bit egyes, $|x|$ értéke $= \bar{x}$ komplementense $+1$.

-1 alakja $11 \dots 11_2$.

Kettes komplement számábrázolás

n -biten: előjelbites számábrázolást keresünk, ahol nincs $+0$ és -0 .

$$\bar{x} = \begin{cases} x & \text{ha } x \text{ nem negatív,} \\ 2^n - |x| & \text{ha } x \text{ negatív.} \end{cases}$$

$2^n - |x|$ kiszámítása n -bites szavak közti bitműveletekkel: $|x|$ bitenkénti komplemente $+1$, ugyanis

$2^n - |x| = (2^n - 1) - |x| + 1 = 11 \dots 1_2 - |x| + 1$. Mivel $|x| = 2^n - (2^n - |x|)$, ezért x értékének meghatározása $\bar{x}$ -ből ugyanígy történik, azaz ha az első bit egyes, $|x|$ értéke $= \bar{x}$ komplemente $+1$.

-1 alakja $11 \dots 11_2$. -2 alakja

Kettes komplementum számábrázolás

n -biten: előjelbites számábrázolást keresünk, ahol nincs $+0$ és -0 .

$$\bar{x} = \begin{cases} x & \text{ha } x \text{ nem negatív,} \\ 2^n - |x| & \text{ha } x \text{ negatív.} \end{cases}$$

$2^n - |x|$ kiszámítása n -bites szavak közti bitműveletekkel: $|x|$ bitenkénti komplementum + 1, ugyanis

$2^n - |x| = (2^n - 1) - |x| + 1 = 11 \dots 1_2 - |x| + 1$. Mivel $|x| = 2^n - (2^n - |x|)$, ezért x értékének meghatározása $\bar{x}$ -ből ugyanígy történik, azaz ha az első bit egyes, $|x|$ értéke $= \bar{x}$ komplementum + 1.

-1 alakja $11 \dots 11_2$. -2 alakja $11 \dots 10_2$.

Kettes komplementum számábrázolás

n -biten: előjelbites számábrázolást keresünk, ahol nincs $+0$ és -0 .

$$\bar{x} = \begin{cases} x & \text{ha } x \text{ nem negatív,} \\ 2^n - |x| & \text{ha } x \text{ negatív.} \end{cases}$$

$2^n - |x|$ kiszámítása n -bites szavak közti bitműveletekkel: $|x|$ bitenkénti komplementum + 1, ugyanis

$2^n - |x| = (2^n - 1) - |x| + 1 = 11 \dots 1_2 - |x| + 1$. Mivel $|x| = 2^n - (2^n - |x|)$, ezért x értékének meghatározása $\bar{x}$ -ből ugyanígy történik, azaz ha az első bit egyes, $|x|$ értéke $= \bar{x}$ komplementum + 1.

-1 alakja $11 \dots 11_2$. -2 alakja $11 \dots 10_2$. -3 alakja

Kettes komplementens számábrázolás

n -biten: előjelbites számábrázolást keresünk, ahol nincs $+0$ és -0 .

$$\bar{x} = \begin{cases} x & \text{ha } x \text{ nem negatív,} \\ 2^n - |x| & \text{ha } x \text{ negatív.} \end{cases}$$

$2^n - |x|$ kiszámítása n -bites szavak közti bitműveletekkel: $|x|$ bitenkénti komplementese $+1$, ugyanis

$2^n - |x| = (2^n - 1) - |x| + 1 = 11 \dots 1_2 - |x| + 1$. Mivel $|x| = 2^n - (2^n - |x|)$, ezért x értékének meghatározása $\bar{x}$ -ből ugyanígy történik, azaz ha az első bit egyes, $|x|$ értéke $= \bar{x}$ komplementese $+1$.

-1 alakja $11 \dots 11_2$. -2 alakja $11 \dots 10_2$. -3 alakja $11 \dots 01_2$.

Kettes komplementens számábrázolás

n -biten: előjelbites számábrázolást keresünk, ahol nincs $+0$ és -0 .

$$\bar{x} = \begin{cases} x & \text{ha } x \text{ nem negatív,} \\ 2^n - |x| & \text{ha } x \text{ negatív.} \end{cases}$$

$2^n - |x|$ kiszámítása n -bites szavak közti bitműveletekkel: $|x|$ bitenkénti komplementese $+1$, ugyanis

$2^n - |x| = (2^n - 1) - |x| + 1 = 11 \dots 1_2 - |x| + 1$. Mivel $|x| = 2^n - (2^n - |x|)$, ezért x értékének meghatározása $\bar{x}$ -ből ugyanígy történik, azaz ha az első bit egyes, $|x|$ értéke $= \bar{x}$ komplementese $+1$.

-1 alakja $11 \dots 11_2$. -2 alakja $11 \dots 10_2$. -3 alakja $11 \dots 01_2$.

Példa

legyen $n = 4$, $x = -5$: $-5 \rightarrow$

Kettes komplementens számábrázolás

n -biten: előjelbites számábrázolást keresünk, ahol nincs $+0$ és -0 .

$$\bar{x} = \begin{cases} x & \text{ha } x \text{ nem negatív,} \\ 2^n - |x| & \text{ha } x \text{ negatív.} \end{cases}$$

$2^n - |x|$ kiszámítása n -bites szavak közti bitműveletekkel: $|x|$ bitenkénti komplementese $+1$, ugyanis

$2^n - |x| = (2^n - 1) - |x| + 1 = 11 \dots 1_2 - |x| + 1$. Mivel $|x| = 2^n - (2^n - |x|)$, ezért x értékének meghatározása $\bar{x}$ -ből ugyanígy történik, azaz ha az első bit egyes, $|x|$ értéke $= \bar{x}$ komplementese $+1$.

-1 alakja $11 \dots 11_2$. -2 alakja $11 \dots 10_2$. -3 alakja $11 \dots 01_2$.

Példa

legyen $n = 4$, $x = -5$: $-5 \rightarrow \bar{x} = 16 - 5$

Kettes komplementens számábrázolás

n -biten: előjelbites számábrázolást keresünk, ahol nincs $+0$ és -0 .

$$\bar{x} = \begin{cases} x & \text{ha } x \text{ nem negatív,} \\ 2^n - |x| & \text{ha } x \text{ negatív.} \end{cases}$$

$2^n - |x|$ kiszámítása n -bites szavak közti bitműveletekkel: $|x|$ bitenkénti komplementese $+1$, ugyanis

$2^n - |x| = (2^n - 1) - |x| + 1 = 11 \dots 1_2 - |x| + 1$. Mivel $|x| = 2^n - (2^n - |x|)$, ezért x értékének meghatározása $\bar{x}$ -ből ugyanígy történik, azaz ha az első bit egyes, $|x|$ értéke $= \bar{x}$ komplementese $+1$.

-1 alakja $11 \dots 11_2$. -2 alakja $11 \dots 10_2$. -3 alakja $11 \dots 01_2$.

Példa

legyen $n = 4$, $x = -5$: $-5 \rightarrow \bar{x} = 16 - 5 = 11$

Kettes komplementens számábrázolás

n -biten: előjelbites számábrázolást keresünk, ahol nincs $+0$ és -0 .

$$\bar{x} = \begin{cases} x & \text{ha } x \text{ nem negatív,} \\ 2^n - |x| & \text{ha } x \text{ negatív.} \end{cases}$$

$2^n - |x|$ kiszámítása n -bites szavak közti bitműveletekkel: $|x|$ bitenkénti komplementese $+1$, ugyanis

$2^n - |x| = (2^n - 1) - |x| + 1 = 11 \dots 1_2 - |x| + 1$. Mivel $|x| = 2^n - (2^n - |x|)$, ezért x értékének meghatározása $\bar{x}$ -ből ugyanígy történik, azaz ha az első bit egyes, $|x|$ értéke $= \bar{x}$ komplementese $+1$.

-1 alakja $11 \dots 11_2$. -2 alakja $11 \dots 10_2$. -3 alakja $11 \dots 01_2$.

Példa

legyen $n = 4$, $x = -5$: $-5 \rightarrow \bar{x} = 16 - 5 = 11 = 1011_2$

Kettes komplementum számábrázolás

n -biten: előjelbites számábrázolást keresünk, ahol nincs $+0$ és -0 .

$$\bar{x} = \begin{cases} x & \text{ha } x \text{ nem negatív,} \\ 2^n - |x| & \text{ha } x \text{ negatív.} \end{cases}$$

$2^n - |x|$ kiszámítása n -bites szavak közti bitműveletekkel: $|x|$ bitenkénti komplementum + 1, ugyanis

$2^n - |x| = (2^n - 1) - |x| + 1 = 11 \dots 1_2 - |x| + 1$. Mivel $|x| = 2^n - (2^n - |x|)$, ezért x értékének meghatározása $\bar{x}$ -ből ugyanígy történik, azaz ha az első bit egyes, $|x|$ értéke = $\bar{x}$ komplementum + 1.

-1 alakja $11 \dots 11_2$. -2 alakja $11 \dots 10_2$. -3 alakja $11 \dots 01_2$.

Példa

legyen $n = 4$, $x = -5$: $-5 \rightarrow \bar{x} = 16 - 5 = 11 = 1011_2$
bitműveletekkel:

$$x = -5 \rightarrow |x| = 5$$

Kettes komplement számábrázolás

***n*-biten:** előjelbites számábrázolást keresünk, ahol nincs $+0$ és -0 .

$$\bar{x} = \begin{cases} x & \text{ha } x \text{ nem negatív,} \\ 2^n - |x| & \text{ha } x \text{ negatív.} \end{cases}$$

$2^n - |x|$ kiszámítása n -bites szavak közti bitműveletekkel: $|x|$ bitenkénti komplemente $+1$, ugyanis

$2^n - |x| = (2^n - 1) - |x| + 1 = 11 \dots 1_2 - |x| + 1$. Mivel $|x| = 2^n - (2^n - |x|)$, ezért x értékének meghatározása $\bar{x}$ -ből ugyanígy történik, azaz ha az első bit egyes, $|x|$ értéke $= \bar{x}$ komplemente $+1$.

-1 alakja $11 \dots 11_2$. -2 alakja $11 \dots 10_2$. -3 alakja $11 \dots 01_2$.

Példa

legyen $n = 4$, $x = -5$: $-5 \rightarrow \bar{x} = 16 - 5 = 11 = 1011_2$
bitműveletekkel:

$x = -5 \rightarrow |x| = 5 \rightarrow 0101_2$

Kettes komplementum számábrázolás

n -biten: előjelbites számábrázolást keresünk, ahol nincs $+0$ és -0 .

$$\bar{x} = \begin{cases} x & \text{ha } x \text{ nem negatív,} \\ 2^n - |x| & \text{ha } x \text{ negatív.} \end{cases}$$

$2^n - |x|$ kiszámítása n -bites szavak közti bitműveletekkel: $|x|$ bitenkénti komplementum + 1, ugyanis

$2^n - |x| = (2^n - 1) - |x| + 1 = 11 \dots 1_2 - |x| + 1$. Mivel $|x| = 2^n - (2^n - |x|)$, ezért x értékének meghatározása $\bar{x}$ -ből ugyanígy történik, azaz ha az első bit egyes, $|x|$ értéke = $\bar{x}$ komplementum + 1.

-1 alakja $11 \dots 11_2$. -2 alakja $11 \dots 10_2$. -3 alakja $11 \dots 01_2$.

Példa

legyen $n = 4$, $x = -5$: $-5 \rightarrow \bar{x} = 16 - 5 = 11 = 1011_2$
bitműveletekkel:

$x = -5 \rightarrow |x| = 5 \rightarrow 0101_2 \rightarrow \bar{x} = 1010_2 + 1_2 = 1011_2$

Kettes komplementum számábrázolás

n -biten: előjelbites számábrázolást keresünk, ahol nincs $+0$ és -0 .

$$\bar{x} = \begin{cases} x & \text{ha } x \text{ nem negatív,} \\ 2^n - |x| & \text{ha } x \text{ negatív.} \end{cases}$$

$2^n - |x|$ kiszámítása n -bites szavak közti bitműveletekkel: $|x|$ bitenkénti komplementum + 1, ugyanis

$2^n - |x| = (2^n - 1) - |x| + 1 = 11\dots 1_2 - |x| + 1$. Mivel $|x| = 2^n - (2^n - |x|)$, ezért x értékének meghatározása $\bar{x}$ -ből ugyanígy történik, azaz ha az első bit egyes, $|x|$ értéke = $\bar{x}$ komplementum + 1.

-1 alakja $11\dots 11_2$. -2 alakja $11\dots 10_2$. -3 alakja $11\dots 01_2$.

Példa

legyen $n = 4$, $x = -5$: $-5 \rightarrow \bar{x} = 16 - 5 = 11 = 1011_2$
bitműveletekkel:

$x = -5 \rightarrow |x| = 5 \rightarrow 0101_2 \rightarrow \bar{x} = 1010_2 + 1_2 = 1011_2$

Visszaalakítás: $\bar{x} = 1011_2$

Kettes komplementum számábrázolás

n -biten: előjelbites számábrázolást keresünk, ahol nincs $+0$ és -0 .

$$\bar{x} = \begin{cases} x & \text{ha } x \text{ nem negatív,} \\ 2^n - |x| & \text{ha } x \text{ negatív.} \end{cases}$$

$2^n - |x|$ kiszámítása n -bites szavak közti bitműveletekkel: $|x|$ bitenkénti komplementum + 1, ugyanis

$2^n - |x| = (2^n - 1) - |x| + 1 = 11 \dots 1_2 - |x| + 1$. Mivel $|x| = 2^n - (2^n - |x|)$, ezért x értékének meghatározása $\bar{x}$ -ből ugyanígy történik, azaz ha az első bit egyes, $|x|$ értéke = $\bar{x}$ komplementum + 1.

-1 alakja $11 \dots 11_2$. -2 alakja $11 \dots 10_2$. -3 alakja $11 \dots 01_2$.

Példa

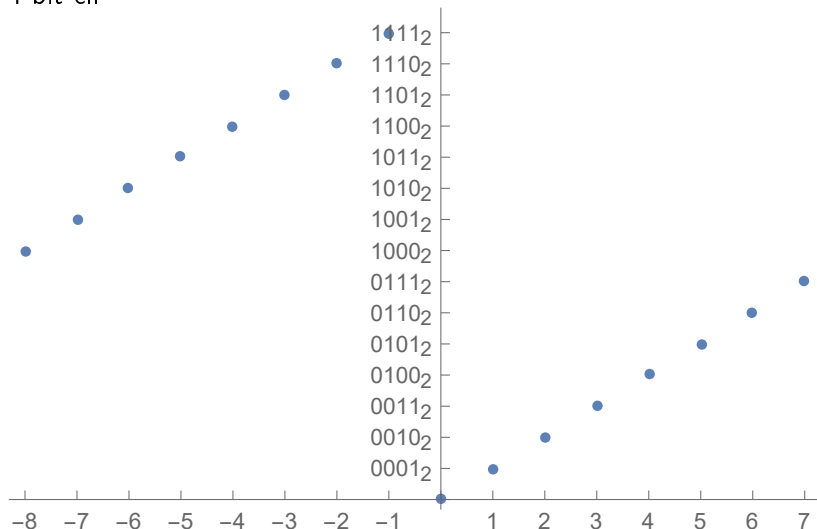
legyen $n = 4$, $x = -5$: $-5 \rightarrow \bar{x} = 16 - 5 = 11 = 1011_2$
bitműveletekkel:

$x = -5 \rightarrow |x| = 5 \rightarrow 0101_2 \rightarrow \bar{x} = 1010_2 + 1_2 = 1011_2$

Visszaalakítás: $\bar{x} = 1011_2 \rightarrow x = 0100_2 + 1_2 = 0101_2 = 5$.

Kettes komplement

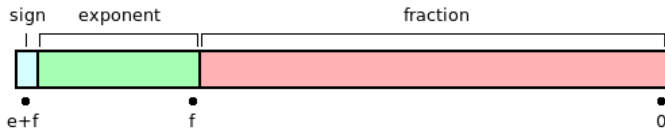
4 bit-en



IEEE 754-2008, ISO/IEC/IEEE 60559:2011

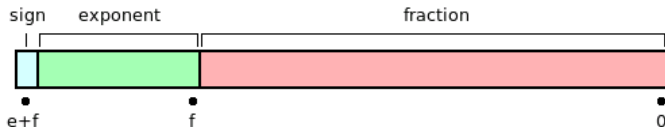
Lebegőpontos számábrázolás

IEEE 754-2008, ISO/IEC/IEEE 60559:2011



Lebegőpontos számábrázolás

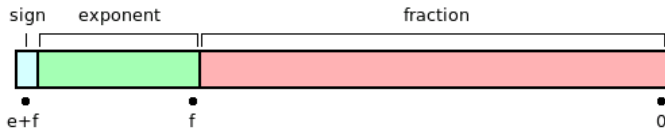
IEEE 754-2008, ISO/IEC/IEEE 60559:2011



	s =előjel	e =kitevő	mantissza	összesen	eltolás (bias)
szimpla	1	8	23	32	127 (01111111)
dupla	1	11	52	64	1023 (0111111111)

Lebegőpontos számábrázolás

IEEE 754-2008, ISO/IEC/IEEE 60559:2011

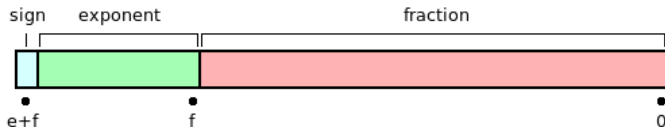


	s=előjel	e=kitevő	mantissza	összesen	eltolás (bias)
szimpla	1	8	23	32	127 (01111111)
dupla	1	11	52	64	1023 (0111111111)
szimpla:					

$$(-1)^s (1.b_{22}b_{21} \dots b_0)_2 \cdot 2^{e-127} = \left(1 + \sum_{i=1}^{23} b_{23-i} 2^{-i} \right) \cdot 2^{e-127}$$

Lebegőpontos számábrázolás

IEEE 754-2008, ISO/IEC/IEEE 60559:2011



	s=előjel	e=kitevő	mantissza	összesen	eltolás (bias)
szimpla	1	8	23	32	127 (01111111)
dupla	1	11	52	64	1023 (0111111111)
szimpla:					

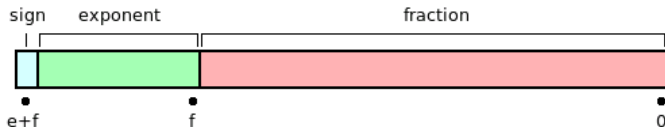
$$(-1)^s (1.b_{22}b_{21} \dots b_0)_2 \cdot 2^{e-127} = \left(1 + \sum_{i=1}^{23} b_{23-i} 2^{-i} \right) \cdot 2^{e-127}$$

dupla:

$$(-1)^s (1.b_{51}b_{50} \dots b_0)_2 \cdot 2^{e-1023} = \left(1 + \sum_{i=1}^{52} b_{52-i} 2^{-i} \right) \cdot 2^{e-1023}$$

Lebegőpontos számábrázolás

IEEE 754-2008, ISO/IEC/IEEE 60559:2011



	s=előjel	e=kitevő	mantissza	összesen	eltolás (bias)
szimpla	1	8	23	32	127 (01111111)
dupla	1	11	52	64	1023 (0111111111)
szimpla:					

$$(-1)^s (1.b_{22}b_{21} \dots b_0)_2 \cdot 2^{e-127} = \left(1 + \sum_{i=1}^{23} b_{23-i} 2^{-i} \right) \cdot 2^{e-127}$$

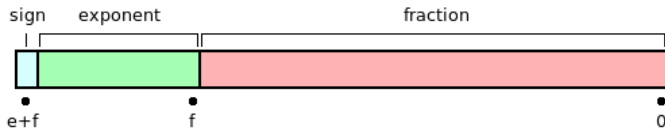
dupla:

$$(-1)^s (1.b_{51}b_{50} \dots b_0)_2 \cdot 2^{e-1023} = \left(1 + \sum_{i=1}^{52} b_{52-i} 2^{-i} \right) \cdot 2^{e-1023}$$

Például dupla pontosság esetén $2^{52} = 4\,503\,599\,627\,370\,496$ és $2^{53} = 9\,007\,199\,254\,740\,992$ között csak az egészek vannak

Lebegőpontos számábrázolás

IEEE 754-2008, ISO/IEC/IEEE 60559:2011



	s=előjel	e=kitevő	mantissza	összesen	eltolás (bias)
szimpla	1	8	23	32	127 (01111111)
dupla	1	11	52	64	1023 (0111111111)
szimpla:					

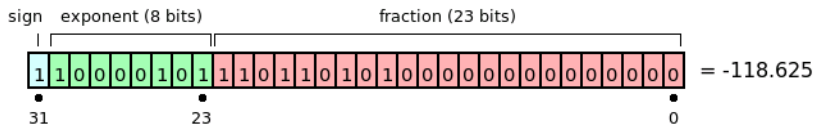
$$(-1)^s (1.b_{22}b_{21} \dots b_0)_2 \cdot 2^{e-127} = \left(1 + \sum_{i=1}^{23} b_{23-i} 2^{-i} \right) \cdot 2^{e-127}$$

dupla:

$$(-1)^s (1.b_{51}b_{50} \dots b_0)_2 \cdot 2^{e-1023} = \left(1 + \sum_{i=1}^{52} b_{52-i} 2^{-i} \right) \cdot 2^{e-1023}$$

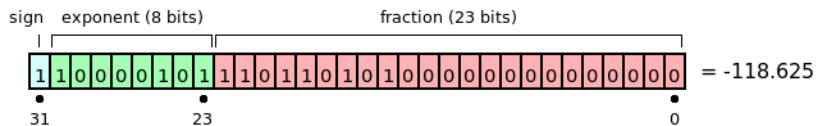
Például dupla pontosság esetén $2^{52} = 4\,503\,599\,627\,370\,496$ és $2^{53} = 9\,007\,199\,254\,740\,992$ között csak az egészek vannak

Példa I.



előjel (sign) 1 $\rightarrow$ negatív

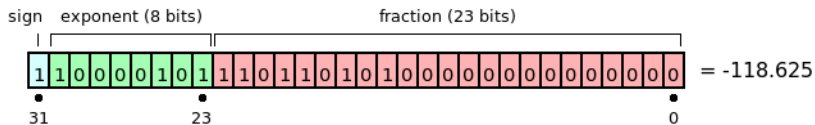
Példa I.



előjel (sign) 1 $\rightarrow$ negatív

kitevő (exponent) $10000101_2 - 01111111_2 = 00000110_2$, azaz 6

Példa I.

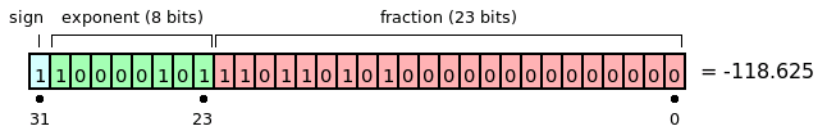


előjel (sign) 1 $\rightarrow$ negatív

kitevő (exponent) $10000101_2 - 01111111_2 = 00000110_2$, azaz 6

mantissa (1.significand) 1.110110101_2 ,

Példa I.



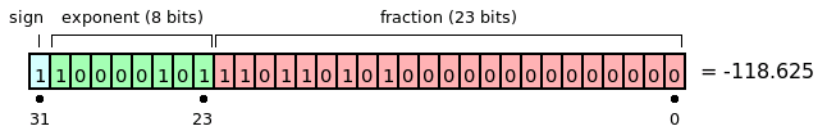
előjel (sign) 1 $\rightarrow$ negatív

kitevő (exponent) $10000101_2 - 01111111_2 = 00000110_2$, azaz 6

mantissa (1.significand) 1.110110101_2 ,

így a szám -1110110.101_2 ,

Példa I.



előjel (sign) 1 $\rightarrow$ negatív

kitevő (exponent) $10000101_2 - 01111111_2 = 00000110_2$, azaz 6

mantissa (1.significand) 1.110110101_2 ,

így a szám -1110110.101_2 , azaz -118.625

Példa

Írjuk fel -14.3 IEEE 754 szabvány szerinti ábrázolását 32 biten.

Példa

Írjuk fel -14.3 IEEE 754 szabvány szerinti ábrázolását 32 biten.

Megoldás: $14 = 1110_2$, $0.3 = 0.01001\dots_2$, tehát a tört első 24 jegye $1110.01001100110011001100$, a kitevő 3, az első 1-est leahagyva a *fraction* rész 23 bitje 11001001100110011001100 .

Példa

Írjuk fel -14.3 IEEE 754 szabvány szerinti ábrázolását 32 biten.

Megoldás: $14 = 1110_2$, $0.3 = 0.01001\dots_2$, tehát a tört első 24 jegye $1110.01001100110011001100$, a kitevő 3, az első 1-est leahagyva a *fraction* rész 23 bitje 11001001100110011001100 . A kitevőhöz 127-et adva: $11 + 01111111 = 10000010$, tehát a szám ábrázolása az utolsó jegy **kerekítése** után:

Példa

Írjuk fel -14.3 IEEE 754 szabvány szerinti ábrázolását 32 biten.

Megoldás: $14 = 1110_2$, $0.3 = 0.01001\dots_2$, tehát a tört első 24 jegye $1110.01001100110011001100$, a kitevő 3, az első 1-est leahagyva a *fraction* rész 23 bitje 11001001100110011001100 . A kitevőhöz 127-et adva: $11 + 01111111 = 10000010$, tehát a szám ábrázolása az utolsó jegy **kerekítése** után:

1

Példa

Írjuk fel -14.3 IEEE 754 szabvány szerinti ábrázolását 32 biten.

Megoldás: $14 = 1110_2$, $0.3 = 0.01001\dots_2$, tehát a tört első 24 jegye $1110.01001100110011001100$, a kitevő 3, az első 1-est leahagyva a *fraction* rész 23 bitje 11001001100110011001100 . A kitevőhöz 127-et adva: $11 + 01111111 = 10000010$, tehát a szám ábrázolása az utolsó jegy **kerekítése** után:

110000010

Példa

Írjuk fel -14.3 IEEE 754 szabvány szerinti ábrázolását 32 biten.

Megoldás: $14 = 1110_2$, $0.3 = 0.01001\dots_2$, tehát a tört első 24 jegye $1110.01001100110011001100$, a kitevő 3, az első 1-est leahagyva a *fraction* rész 23 bitje 11001001100110011001100 . A kitevőhöz 127-et adva: $11 + 01111111 = 10000010$, tehát a szám ábrázolása az utolsó jegy **kerekítése** után:

11000001011001001100110011001101

Példa

Írjuk fel -14.3 IEEE 754 szabvány szerinti ábrázolását 32 biten.

Megoldás: $14 = 1110_2$, $0.3 = 0.01001\dots_2$, tehát a tört első 24 jegye $1110.01001100110011001100$, a kitevő 3, az első 1-est leahagyva a *fraction* rész 23 bitje 11001001100110011001100 . A kitevőhöz 127-et adva: $11 + 01111111 = 10000010$, tehát a szám ábrázolása az utolsó jegy **kerekítése** után:

11000001011001001100110011001101

Kerekítés: ha az első elhagyott bit 1, akkor a mantissza utolsó bitjéhez 1-et adunk!

Az Interneten számtalan konverteren tesztelhetők a konverziók, pl. ez az IEEE754-re vagy ez több különböző típusra.

ASCII kódtábla

ASCII – American Standard Code for Information Interchange

0	00		<control>	59	3B	;	SEMICOLON
...				60	3C	<	LESS-THAN SIGN
31	1F		<control>	61	3D	=	EQUALS SIGN
32	20		SPACE	62	3E	>	GREATER-THAN SIGN
33	21	!	EXCLAMATION MARK	63	3F	?	QUESTION MARK
34	22	"	QUOTATION MARK	64	40	@	COMMERCIAL AT
35	23	#	NUMBER SIGN	65	41	A	LATIN CAPITAL LETTER A
36	24	\$	DOLLAR SIGN	...			
37	25	%	PERCENT SIGN	90	5A	Z	LATIN CAPITAL LETTER Z
38	26	&	AMPERSAND	91	5B	[	LEFT SQUARE BRACKET
39	27	'	APOSTROPHE	92	5C	\	REVERSE SOLIDUS
40	28	(	LEFT PARENTHESIS	93	5D	]	RIGHT SQUARE BRACKET
41	29	)	RIGHT PARENTHESIS	94	5E	^	CIRCUMFLEX ACCENT
42	2A	*	ASTERISK	95	5F	_	LOW LINE
43	2B	+	PLUS SIGN	96	60	`	GRAVE ACCENT
44	2C	,	COMMA	97	61	a	LATIN SMALL LETTER A
45	2D	-	HYPHEN-MINUS	...			
46	2E	.	FULL STOP	122	7A	z	LATIN SMALL LETTER Z
47	2F	/	SOLIDUS	123	7B	{	LEFT CURLY BRACKET
48	30	0	DIGIT ZERO	124	7C		VERTICAL LINE
...				125	7D	}	RIGHT CURLY BRACKET
57	39	9	DIGIT NINE	126	7E	~	TILDE
58	3A	:	COLON	127	7F	<control>	<control>

Már majdnem csak történelem

- ❶ ISO-8859-1 Latin1 (West European)
- ❷ ISO-8859-2 Latin2 (East European)
- ❸ ISO-8859-3 Latin3 (South European)
- ❹ ISO-8859-4 Latin4 (North European)
- ❺ ISO-8859-5 Cyrillic
- ❻ ISO-8859-6 Arabic
- ❼ ISO-8859-7 Greek
- ❽ ISO-8859-8 Hebrew
- ❾ ISO-8859-9 Latin5 (Turkish)
- ❿ ISO-8859-10 Latin6 (Nordic)

Már majdnem csak történelem

ISO-8859-2, Microsoft CP1250 (Windows Latin2), CP852
(DOSLatin2)

ISO-8859-1	C1	Á	U+00C1	LATIN CAPITAL LETTER A WITH ACUTE
ISO-8859-1	E1	á	U+00E1	LATIN SMALL LETTER A WITH ACUTE

Már majdnem csak történelem

ISO-8859-2, Microsoft CP1250 (Windows Latin2), CP852
(DOSLatin2)

ISO-8859-1	C1	Á	U+00C1	LATIN CAPITAL LETTER A WITH ACUTE
ISO-8859-1	E1	á	U+00E1	LATIN SMALL LETTER A WITH ACUTE
ISO-8859-1	D5	Ő	U+00D5	LATIN CAPITAL LETTER O WITH TILDE
ISO-8859-1	DB	Ű	U+00DB	LATIN CAPITAL LETTER U WITH CIRCUMFLEX
ISO-8859-1	F5	ő	U+00F5	LATIN SMALL LETTER O WITH TILDE
ISO-8859-1	FB	ű	U+00FB	LATIN SMALL LETTER U WITH CIRCUMFLEX

Már majdnem csak történelem

ISO-8859-2, Microsoft CP1250 (Windows Latin2), CP852 (DOSLatin2)

ISO-8859-1	C1	Á	U+00C1	LATIN CAPITAL LETTER A WITH ACUTE
ISO-8859-1	E1	á	U+00E1	LATIN SMALL LETTER A WITH ACUTE
ISO-8859-1	D5	Ō	U+00D5	LATIN CAPITAL LETTER O WITH TILDE
ISO-8859-1	DB	Ū	U+00DB	LATIN CAPITAL LETTER U WITH CIRCUMFLEX
ISO-8859-1	F5	ō	U+00F5	LATIN SMALL LETTER O WITH TILDE
ISO-8859-1	FB	ū	U+00FB	LATIN SMALL LETTER U WITH CIRCUMFLEX
ISO-8859-2	D5	Ō	U+0150	LATIN CAPITAL LETTER O WITH DOUBLE ACUTE
ISO-8859-2	DB	Ū	U+0170	LATIN CAPITAL LETTER U WITH DOUBLE ACUTE
ISO-8859-2	F5	ō	U+0151	LATIN SMALL LETTER O WITH DOUBLE ACUTE
ISO-8859-2	FB	ū	U+0171	LATIN SMALL LETTER U WITH DOUBLE ACUTE

Már majdnem csak történelem

ISO-8859-2, Microsoft CP1250 (Windows Latin2), CP852 (DOSLatin2)

ISO-8859-1	C1	Ã	U+00C1	LATIN CAPITAL LETTER A WITH ACUTE
ISO-8859-1	E1	á	U+00E1	LATIN SMALL LETTER A WITH ACUTE
ISO-8859-1	D5	Õ	U+00D5	LATIN CAPITAL LETTER O WITH TILDE
ISO-8859-1	DB	Ô	U+00DB	LATIN CAPITAL LETTER U WITH CIRCUMFLEX
ISO-8859-1	F5	õ	U+00F5	LATIN SMALL LETTER O WITH TILDE
ISO-8859-1	FB	ô	U+00FB	LATIN SMALL LETTER U WITH CIRCUMFLEX
ISO-8859-2	D5	Ō	U+0150	LATIN CAPITAL LETTER O WITH DOUBLE ACUTE
ISO-8859-2	DB	Ū	U+0170	LATIN CAPITAL LETTER U WITH DOUBLE ACUTE
ISO-8859-2	F5	ō	U+0151	LATIN SMALL LETTER O WITH DOUBLE ACUTE
ISO-8859-2	FB	ū	U+0171	LATIN SMALL LETTER U WITH DOUBLE ACUTE
CP1250	82	,	U+201A	SINGLE LOW-9 QUOTATION MARK
CP1250	84	„	U+201E	DOUBLE LOW-9 QUOTATION MARK
CP1250	85	...	U+2026	HORIZONTAL ELLIPSIS
CP1250	91	‘	U+2018	LEFT SINGLE QUOTATION MARK
CP1250	92	’	U+2019	RIGHT SINGLE QUOTATION MARK
CP1250	93	“	U+201C	LEFT DOUBLE QUOTATION MARK
CP1250	94	”	U+201D	RIGHT DOUBLE QUOTATION MARK
CP1250	96	–	U+2013	EN DASH
CP1250	97	—	U+2014	EM DASH

- U+0000 - U+007F ASCII

Unicode

- U+0000 - U+007F ASCII
- U+0080 - U+00FF Latin-1

- U+0000 - U+007F ASCII
- U+0080 - U+00FF Latin-1
- U+0100 - U+017F Latin Extended-A (latin1, magyar ő, ű)

- U+0000 - U+007F ASCII
- U+0080 - U+00FF Latin-1
- U+0100 - U+017F Latin Extended-A (latin1, magyar ő, ű)
- U+0180 - U+024F Latin Extended-B

- U+0000 - U+007F ASCII
- U+0080 - U+00FF Latin-1
- U+0100 - U+017F Latin Extended-A (latin1, magyar ő, ű)
- U+0180 - U+024F Latin Extended-B
- U+1E00 - U+1EFF Latin Extended Additional

UTF – Unicode Transformation Format

- UTF-8 minden karakter kódja 8, 16, 24 vagy 32-bites.

UTF – Unicode Transformation Format

- UTF-8 minden karakter kódja 8, 16, 24 vagy 32-bites.
- UTF-16 minden karakter kódja 16 vagy 32-bites.

UTF – Unicode Transformation Format

- UTF-8 minden karakter kódja 8, 16, 24 vagy 32-bites.
- UTF-16 minden karakter kódja 16 vagy 32-bites.
- UTF-32 minden karakter 32-bites.

Unicode		UTF-8	a karakter hivatalos neve
U+0020		20	SPACE
U+0030	0	30	DIGIT ZERO
U+0040	@	40	COMMERCIAL AT
U+0041	A	41	LATIN CAPITAL LETTER A
U+0061	a	61	LATIN SMALL LETTER A

Unicode		UTF-8	a karakter hivatalos neve
U+0020		20	SPACE
U+0030	0	30	DIGIT ZERO
U+0040	@	40	COMMERCIAL AT
U+0041	A	41	LATIN CAPITAL LETTER A
U+0061	a	61	LATIN SMALL LETTER A
U+00C1	Á	c3 81	LATIN CAPITAL LETTER A WITH ACUTE
U+00C9	É	c3 89	LATIN CAPITAL LETTER E WITH ACUTE
U+00CD	Í	c3 8d	LATIN CAPITAL LETTER I WITH ACUTE
U+00D3	Ó	c3 93	LATIN CAPITAL LETTER O WITH ACUTE
U+00D6	Ö	c3 96	LATIN CAPITAL LETTER O WITH DIAERESIS
U+00DA	Ú	c3 9a	LATIN CAPITAL LETTER U WITH ACUTE
U+00DC	Ü	c3 9c	LATIN CAPITAL LETTER U WITH DIAERESIS
U+00E1	á	c3 a1	LATIN SMALL LETTER A WITH ACUTE
U+00E9	é	c3 a9	LATIN SMALL LETTER E WITH ACUTE
U+00ED	í	c3 ad	LATIN SMALL LETTER I WITH ACUTE
U+00F3	ó	c3 b3	LATIN SMALL LETTER O WITH ACUTE
U+00F6	ö	c3 b6	LATIN SMALL LETTER O WITH DIAERESIS
U+00FA	ú	c3 ba	LATIN SMALL LETTER U WITH ACUTE
U+00FC	ü	c3 bc	LATIN SMALL LETTER U WITH DIAERESIS

Unicode		UTF-8	a karakter hivatalos neve
U+0020		20	SPACE
U+0030	0	30	DIGIT ZERO
U+0040	@	40	COMMERCIAL AT
U+0041	A	41	LATIN CAPITAL LETTER A
U+0061	a	61	LATIN SMALL LETTER A
U+00C1	Á	c3 81	LATIN CAPITAL LETTER A WITH ACUTE
U+00C9	É	c3 89	LATIN CAPITAL LETTER E WITH ACUTE
U+00CD	Í	c3 8d	LATIN CAPITAL LETTER I WITH ACUTE
U+00D3	Ó	c3 93	LATIN CAPITAL LETTER O WITH ACUTE
U+00D6	Ö	c3 96	LATIN CAPITAL LETTER O WITH DIAERESIS
U+00DA	Ú	c3 9a	LATIN CAPITAL LETTER U WITH ACUTE
U+00DC	Ü	c3 9c	LATIN CAPITAL LETTER U WITH DIAERESIS
U+00E1	á	c3 a1	LATIN SMALL LETTER A WITH ACUTE
U+00E9	é	c3 a9	LATIN SMALL LETTER E WITH ACUTE
U+00ED	í	c3 ad	LATIN SMALL LETTER I WITH ACUTE
U+00F3	ó	c3 b3	LATIN SMALL LETTER O WITH ACUTE
U+00F6	ö	c3 b6	LATIN SMALL LETTER O WITH DIAERESIS
U+00FA	ú	c3 ba	LATIN SMALL LETTER U WITH ACUTE
U+00FC	ü	c3 bc	LATIN SMALL LETTER U WITH DIAERESIS
U+0150	Ő	c5 90	LATIN CAPITAL LETTER O WITH DOUBLE ACUTE
U+0151	ő	c5 91	LATIN SMALL LETTER O WITH DOUBLE ACUTE

Unicode		UTF-8	a karakter hivatalos neve
U+0020		20	SPACE
U+0030	0	30	DIGIT ZERO
U+0040	@	40	COMMERCIAL AT
U+0041	A	41	LATIN CAPITAL LETTER A
U+0061	a	61	LATIN SMALL LETTER A
U+00C1	Á	c3 81	LATIN CAPITAL LETTER A WITH ACUTE
U+00C9	É	c3 89	LATIN CAPITAL LETTER E WITH ACUTE
U+00CD	Í	c3 8d	LATIN CAPITAL LETTER I WITH ACUTE
U+00D3	Ó	c3 93	LATIN CAPITAL LETTER O WITH ACUTE
U+00D6	Ö	c3 96	LATIN CAPITAL LETTER O WITH DIAERESIS
U+00DA	Ú	c3 9a	LATIN CAPITAL LETTER U WITH ACUTE
U+00DC	Ü	c3 9c	LATIN CAPITAL LETTER U WITH DIAERESIS
U+00E1	á	c3 a1	LATIN SMALL LETTER A WITH ACUTE
U+00E9	é	c3 a9	LATIN SMALL LETTER E WITH ACUTE
U+00ED	í	c3 ad	LATIN SMALL LETTER I WITH ACUTE
U+00F3	ó	c3 b3	LATIN SMALL LETTER O WITH ACUTE
U+00F6	ö	c3 b6	LATIN SMALL LETTER O WITH DIAERESIS
U+00FA	ú	c3 ba	LATIN SMALL LETTER U WITH ACUTE
U+00FC	ü	c3 bc	LATIN SMALL LETTER U WITH DIAERESIS
U+0150	Ő	c5 90	LATIN CAPITAL LETTER O WITH DOUBLE ACUTE
U+0151	ő	c5 91	LATIN SMALL LETTER O WITH DOUBLE ACUTE
U+0170	Ű	c5 b0	LATIN CAPITAL LETTER U WITH DOUBLE ACUTE
U+0171	ű	c5 b1	LATIN SMALL LETTER U WITH DOUBLE ACUTE

UTF8

Kódtartomány (darab)	bináris alak	UTF-8
----------------------	--------------	-------

UTF8

Kódtartomány (darab)	bináris alak	UTF-8
000000-00007F (128)	0zzzzzzz	0zzzzzzz
000080-0007FF (1920)	00000yyy yyzzzzzz	110yyyyy 10zzzzzz
000800-00FFFF (63488)	xxxxyyyy yyzzzzzz	1110xxxx 10yyyyyy 10zzzzzz
010000-10FFFF (1048576)	000wwwxx xxxxyyyy yyzzzzzz	11110www 10xxxxxx 10yyyyyy 10zzzzzz

UTF8

Kódtartomány (darab)	bináris alak	UTF-8
000000-00007F (128)	0zzzzzzz	0zzzzzzz
000080-0007FF (1920)	00000yyy yyzzzzzz	110yyyyy 10zzzzzz
000800-00FFFF (63488)	xxxxyyyy yyzzzzzz	1110xxxx 10yyyyyy 10zzzzzz
010000-10FFFF (1048576)	000wwwxx xxxxyyyy yyzzzzzz	11110www 10xxxxxx 10yyyyyy 10zzzzzz

Á 00C1

UTF8

Kódtartomány (darab)	bináris alak	UTF-8
000000-00007F (128)	0zzzzzzz	0zzzzzzz
000080-0007FF (1920)	00000yyy yyzzzzzz	110yyyyy 10zzzzzz
000800-00FFFF (63488)	xxxxyyyy yyzzzzzz	1110xxxx 10yyyyyy 10zzzzzz
010000-10FFFF (1048576)	000wwwxx xxxxyyyy yyzzzzzz	11110www 10xxxxxx 10yyyyyy 10zzzzzz

Á 00C1→1100 0001

UTF8

Kódtartomány (darab)	bináris alak	UTF-8
000000-00007F (128)	0zzzzzzz	0zzzzzzz
000080-0007FF (1920)	00000yyy yyzzzzzz	110yyyyy 10zzzzzz
000800-00FFFF (63488)	xxxxyyyy yyzzzzzz	1110xxxx 10yyyyyy 10zzzzzz
010000-10FFFF (1048576)	000wwwxx xxxxyyyy yyzzzzzz	11110www 10xxxxxx 10yyyyyy 10zzzzzz

Á 00C1→1100 0001→00011 000001

UTF8

Kódtartomány (darab)	bináris alak	UTF-8
000000-00007F (128)	0zzzzzzz	0zzzzzzz
000080-0007FF (1920)	00000yyy yyzzzzzz	110yyyyy 10zzzzzz
000800-00FFFF (63488)	xxxxyyyy yyzzzzzz	1110xxxx 10yyyyyy 10zzzzzz
010000-10FFFF (1048576)	000wwwxx xxxxyyyy yyzzzzzz	11110www 10xxxxxx 10yyyyyy 10zzzzzz

Á 00C1→1100 0001→00011 000001→11000011 10000001

UTF8

Kódtartomány (darab)	bináris alak	UTF-8
000000-00007F (128)	0zzzzzzz	0zzzzzzz
000080-0007FF (1920)	00000yyy yyzzzzzz	110yyyyy 10zzzzzz
000800-00FFFF (63488)	xxxxyyyy yyzzzzzz	1110xxxx 10yyyyyy 10zzzzzz
010000-10FFFF (1048576)	000wwwxx xxxxyyyy yyzzzzzz	11110www 10xxxxxx 10yyyyyy 10zzzzzz

Á 00C1→1100 0001→00011 000001→11000011 10000001→C3 81

UTF8

Kódtartomány (darab)	bináris alak	UTF-8
000000-00007F (128)	0zzzzzzz	0zzzzzzz
000080-0007FF (1920)	00000yyy yyzzzzzz	110yyyyy 10zzzzzz
000800-00FFFF (63488)	xxxxyyyy yyzzzzzz	1110xxxx 10yyyyyy 10zzzzzz
010000-10FFFF (1048576)	000wwwxx xxxxyyyy yyzzzzzz	11110www 10xxxxxx 10yyyyyy 10zzzzzz

Á 00C1→1100 0001→00011 000001→11000011 10000001→C3 81
Ŕ 00D5→1101 0101→00011 010101→11000011 10010101→C3 95

UTF8

Kódtartomány (darab)	bináris alak	UTF-8
000000-00007F (128)	0zzzzzzz	0zzzzzzz
000080-0007FF (1920)	00000yyy yyzzzzzz	110yyyyy 10zzzzzz
000800-00FFFF (63488)	xxxxyyyy yyzzzzzz	1110xxxx 10yyyyyy 10zzzzzz
010000-10FFFF (1048576)	000wwwxx xxxxyyyy yyzzzzzz	11110www 10xxxxxx 10yyyyyy 10zzzzzz

Á 00C1→1100 0001→00011 000001→11000011 10000001→C3 81
Ŕ 00D5→1101 0101→00011 010101→11000011 10010101→C3 95
Ŗ 0150→0001 0101 0000→00101 010000→11000101
10010000→C5 90

UTF8

Kódtartomány (darab)	bináris alak	UTF-8
000000-00007F (128)	0zzzzzzz	0zzzzzzz
000080-0007FF (1920)	00000yyy yyzzzzzz	110yyyyy 10zzzzzz
000800-00FFFF (63488)	xxxxyyyy yyzzzzzz	1110xxxx 10yyyyyy 10zzzzzz
010000-10FFFF (1048576)	000wwwxx xxxxyyyy yyzzzzzz	11110www 10xxxxxx 10yyyyyy 10zzzzzz

Á 00C1→1100 0001→00011 000001→11000011 10000001→C3 81
Ŕ 00D5→1101 0101→00011 010101→11000011 10010101→C3 95
Ŗ 0150→0001 0101 0000→00101 010000→11000101
10010000→C5 90

Byte Order Mark FEFF

UTF8

Kódtartomány (darab)	bináris alak	UTF-8
000000-00007F (128)	0zzzzzzz	0zzzzzzz
000080-0007FF (1920)	00000yyy yyzzzzzz	110yyyyy 10zzzzzz
000800-00FFFF (63488)	xxxxyyyy yyzzzzzz	1110xxxx 10yyyyyy 10zzzzzz
010000-10FFFF (1048576)	000wwwxx xxxxyyyy yyzzzzzz	11110www 10xxxxxx 10yyyyyy 10zzzzzz

Á 00C1→1100 0001→00011 000001→11000011 10000001→C3 81
Ŕ 00D5→1101 0101→00011 010101→11000011 10010101→C3 95
Ŗ 0150→0001 0101 0000→00101 010000→11000101
10010000→C5 90
Byte Order Mark FEFF→11111110 11111111→
11101111 10111011 10111111

UTF8

Kódtartomány (darab)	bináris alak	UTF-8
000000-00007F (128)	0zzzzzzz	0zzzzzzz
000080-0007FF (1920)	00000yyy yyzzzzzz	110yyyyy 10zzzzzz
000800-00FFFF (63488)	xxxxyyyy yyzzzzzz	1110xxxx 10yyyyyy 10zzzzzz
010000-10FFFF (1048576)	000wwwxx xxxxyyyy yyzzzzzz	11110www 10xxxxxx 10yyyyyy 10zzzzzz

Á 00C1→1100 0001→00011 000001→11000011 10000001→C3 81
Ŕ 00D5→1101 0101→00011 010101→11000011 10010101→C3 95
Ŗ 0150→0001 0101 0000→00101 010000→11000101
10010000→C5 90

Byte Order Mark FEFF→11111110 11111111→
11101111 10111011 10111111→EF BB BF (í»¿ Windows fájlok
elején az UTF-8 formátum jelzése – ha latin-1-ben látjuk)

UTF8

Kódtartomány (darab)	bináris alak	UTF-8
000000-00007F (128)	0zzzzzzz	0zzzzzzz
000080-0007FF (1920)	00000yyy yyzzzzzz	110yyyyy 10zzzzzz
000800-00FFFF (63488)	xxxxyyyy yyzzzzzz	1110xxxx 10yyyyyy 10zzzzzz
010000-10FFFF (1048576)	000wwwxx xxxxyyyy yyzzzzzz	11110www 10xxxxxx 10yyyyyy 10zzzzzz

Á 00C1→1100 0001→00011 000001→11000011 10000001→C3 81
Ŕ 00D5→1101 0101→00011 010101→11000011 10010101→C3 95
Ŗ 0150→0001 0101 0000→00101 010000→11000101
10010000→C5 90

Byte Order Mark FEFF→11111110 11111111→
11101111 10111011 10111111→EF BB BF (í»¿ Windows fájlok
elején az UTF-8 formátum jelzése – ha latin-1-ben látjuk)

Egy file nevű fájl tartalmának hexadecimális módon való
megtekintése parancssorból (hexdump): `hd file`

- 1 Írjuk fel a 13.4 és a -12.6 bináris alakját!
- 2 Írjuk fel -23 és -24 kettes komplementum alakját 8 biten.
- 3 Mennyi az értéke annak a számnak, amelynek kettes komplementum alakja 10101001?
- 4 Mi a -23.4 és a -12.6 IEEE 754-es szabvány szerinti 32-bites alakja?
- 5 Mennyi az értéke a 32-bites lebegőpontos 11000001110101100000000000000000 számnak?
- 6 A magyar ábécé betűit tekintve mi a különbség a Latin-1 és Latin-2 kódolás között?
- 7 Hogyan képződik egy karakter unicode kódjából utf-8-as kódja?
- 8 Az € karakter kódja U+20AC. Írjuk fel binárisan és hexadecimálisan az UTF-8 kódját!