

$$12345 = 5 + 4 \cdot 10 + 3 \cdot 100 + 2 \cdot 1000 + 1 \cdot 10\,000$$

$$\begin{array}{r} 233796 = 6 \cdot 10^0 + 9 \cdot 10 + 7 \cdot 10^2 + 3 \cdot 10^3 \\ + 3 \cdot 10^4 + 2 \cdot 10^5 \end{array}$$

Pomembno! Zakaj smo izbrali ravno 10?

Izberemo lahko katerikoli številko!

Kaj pa če bi vzeli 2?

$$3, 5, 7 \rightarrow 3 \cdot 2^0 + 5 \cdot 2^1 + 7 \cdot 2^2 = 41$$

Naredili smo napako, zaradi katere smo dobili neprimeren rezultat.

Nova konvencija:

→ šteke morajo biti 0 ali 1

→ zapis abcd₍₂₎ to pomeni tisto število v osnovi 2.

$$\begin{array}{r} 1101001_{(2)} = 1 \cdot 2^0 + 0 \cdot 2^1 + 0 \cdot 2^2 \\ + 1 \cdot 2^3 + 0 \cdot 2^4 + 1 \cdot 2^5 + 1 \cdot 2^6 = 105_{(10)} \end{array}$$

Zakaj bi nam bilo mar?

Računalniki lahko enostavno delajo z dvojiškimi predstavitvami

↳ če električna teža po žici, to pomeni 1

↳ sicer to pomeni 0

Nastevanje v bitnem zapisu:

$$1_{(2)}, 10_{(2)}, 11_{(2)}, 100_{(2)}, 101_{(2)}, 110_{(2)}, \dots$$

$$\begin{array}{r} 11011 \\ + \quad \quad 11 \\ \hline 11100 \end{array}$$

$$\begin{array}{r} 56 \\ + \quad 7 \\ \hline 63 \end{array}$$

Algoritem: kako poiščem dvojiški zapis števila?

Je isti kot že en znan algoritem!

while ($a > 0$) { Oglejmo si: delovanje pri $a = 12345$

izračunaj $a \% 10$

$a = a / 10$

$a = 12345$

}

$a \% 10 = 5$

$a = a / 10 \leadsto a = 1234$

}

$a \% 10 = 4$

$a = a / 10 \leadsto a = 123$

}

$a \% 10 = 3$

$a = a / 10 \leadsto a = 12$

}

⋮

Algoritem za izračun dvojiškega zapisa:

while ($a > 0$) {

izračunaj $a \% 2$

$a = a / 2$

$\leadsto$ izračuna zadnje števko dvojiškega zapisa a

$\leadsto$ iz a odstrani zadnje števko

}

obni: izračunane številke

$$123 \cdot 10 = 1230$$

$$123 \cdot 10^7 = 123\ 000\ 000\ 0$$

$$456 / 10 = 45$$

$$456 / 100 = 4$$

$$1101_{(2)} \cdot 10_{(2)}^2 = 11010_{(2)}$$

$$1101_{(2)} / 10_{(2)} = 110_{(2)}$$

$$\text{to get 13 to, but} \\ \ll 1, \gg 1$$

$$2^n$$

$$x \text{ NAND } x = \sim x$$

$$\sim(x \text{ NAND } y) = x \& y$$

$$\sim(\sim x \& \sim y) = x | y$$