

Organizace předmětu, číselné soustavy, pevná řádová čárka a doplňkový kód

ISU-cv01

Ing. Jakub Husa Ph.D.

Vysoké učení technické v Brně, Fakulta informačních technologií
Božetěchova 1/2. 612 66 Brno - Královo Pole
ihusa@fit.vut.cz



14. února 2025

Organizace předmětu

Aby mohl být program vykonán na **procesoru**, musí být nejprve **přeložen** z **programovacího jazyka**, kterému rozumí programátor, do **jazyka symbolických adres (instrukcí)**, lidově zvaného **assembler**, kterému rozumí procesor:

- V předmětu **ISU** se budeme učit programy psát přímo v jazyce procesoru, abychom poznali jak procesor funguje a porozuměli jeho **fyzickým omezením**.

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main()
{
    printf("Hello World!\n");
    return 0;
}
```

gcc -S main.c



```
main:
.LFB2:
    .cfi_startproc
    pushq   %rbp
    .cfi_def_cfa_offset 16
    .cfi_offset 6, -16
    movq    %rsp, %rbp
    .cfi_def_cfa_register 6
    movl    $.LC0, %edi
    call    puts
    movl    $0, %eax
    popq    %rbp
    .cfi_def_cfa 7, 8
    ret
```

Garantem předmětu je doktor **Filip Orság** (orsag@fit.vut.cz):

- Dalšími přednášejícími jsou doktorka **Marta Jaroš**, a doktor **Tomáš Goldmann**.

Výuku cvičení zajišťuje 9 různých vyučujících:

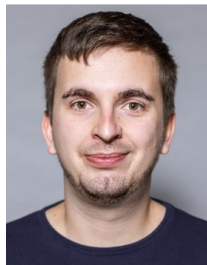
- Vaším vyučujícím jsem já, doktor **Jakub Husa**, konzultace poskytují po dohodě e-mailem (ihusa@fit.vut.cz) v **kanceláři L307**.

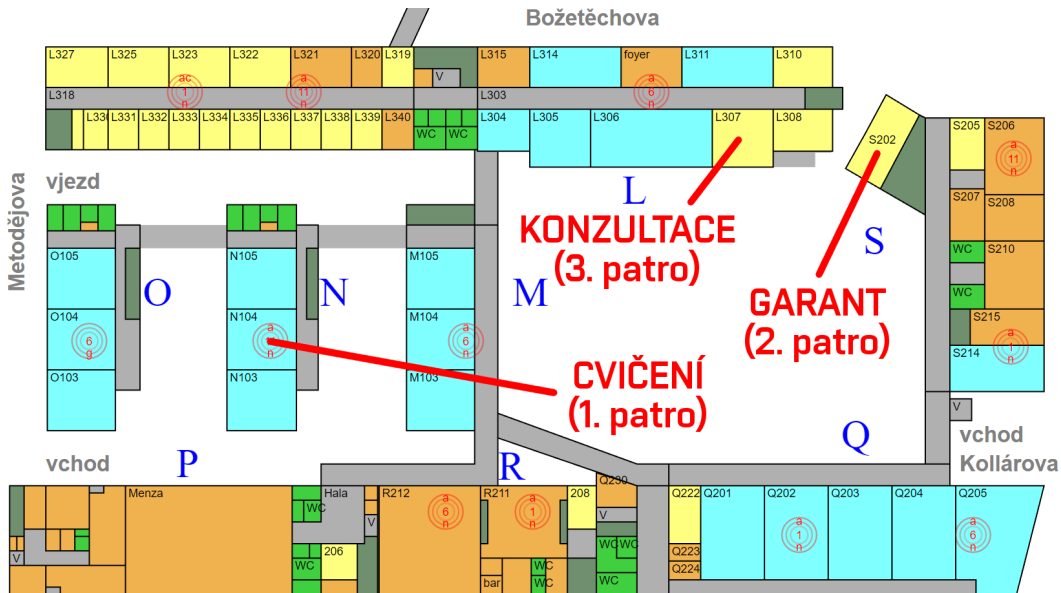
Všechny výukové materiály najdete v **E-learningu**:

- Najdete zde obsah přednášek a cvičení, návody, vývojové nástroje, a spoustu dalších materiálů.
- Sajdy ze cvičení najdete mezi **materiály cvičících**.
- Jako referenční stroj budeme používat **ISU-HUB**.
- Aktuality mohou být zveřejňovány také ve **Studisu**.

Pro oficiální komunikaci vždy používejte váš **školní e-mail**:

- Pro neoficiální komunikaci můžete používat **Discord**.





Během semestru budete psát tři **bodované testy** a **půlsemestrální zkoušku**:

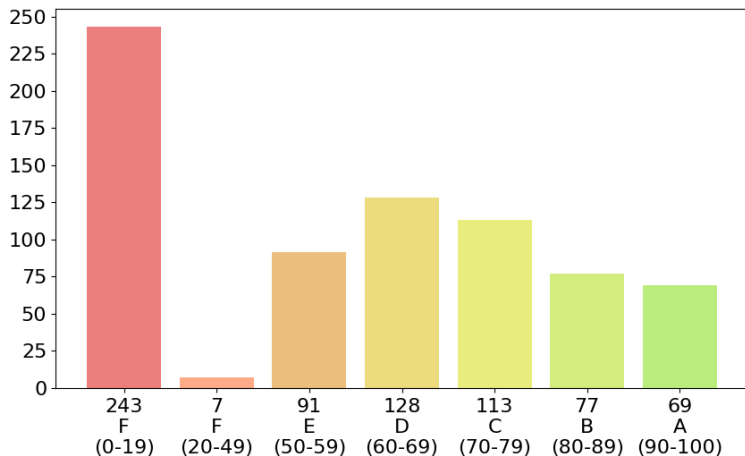
- 6. týden (místo cvičení) – **1. test (6b)**.
- 7. týden (místo přednášky) – **půlsemestrálka (16b)**.
- 10. týden (místo cvičení) – **2. test (9b)**.
- 13. týden (místo cvičení) – **3. test (9b)**.
- Pokud se vám během semestru podaří získat **zápočet** (min **20b** ze 40), na konci semestru budete psát také **semestrální zkoušku** (min **25b** ze 60).

Za aktivitu na cvičeních můžete během semestru získat **bonusové body**:

- Maximálně **1** bod za týden a celkem maximálně **5** bodů za semestr.
- Bonusové body získáváte pouze v případě úspěšného zvládnutí předmětu, **nepočítají** se tedy do zápočtu a **nemohou** vaši známku změnit z **F** na **E**.

Vaše přítomnost ve výuce je **ZCELA DOBROVOLNÁ**:

- Zmeškaná cvičení si můžete nahradit v jiném termínu (Út a St 14:00–20:00, Pá 14:00–18:00, vždy v **N104**), a po dohodě e-mailem i u jiného cvičícího.
- Bodované testy se píší elektronicky a **můžete** přijít pouze na **zapsaný termín**!



Minulý rok předmět ISU nezvládlo 34.3% studentů,
97.2% z nich proto že **nezískali zápočet!**

Úmrtnost studentů
v 1. semestru (2024):

- IDM – 39.2%
- IEL – 28.6%
- ILG – 26.9%
- IUS – 39.6%
- IZP – 28.4%

Úmrtnost studentů
v 2. semestru (2024):

- IMA1 – 50.8%
- INC – 23.7%
- IOS – 41.3%
- ISU – 34.3%
- IZLO – 25.5%

Znalosti z ISU budete potřebovat i u **státní závěrečné zkoušky** – otázky 9, 18 a 41!

Číselné soustavy

Čísla obvykle reprezentujeme jako posloupnost číslic v poziční číselné soustavě s nějakým základem:

- Nejčastěji používáme desítkovou soustavu se základem 10 a číslicemi {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}.
- Na strojové úrovni se obvykle používá soustava dvojková, šestnáctková, a výjimečně i osmičková.

V matematice základ soustavy označujeme závorkami s pravým dolním indexem:

$$(1000001)_2 = (101)_8 = (65)_{10} = (41)_{16}$$

V programování základ soustavy označujeme předponou na začátku nebo příponou na konci čísla:

$$0b1000001 = 0o101 = 65 = 0x41$$

$$1000001b = 101o = 65 = 41h$$

Zápis čísel v různých soustavách

	2	8	10	16
0	0	0	0	0
1	1	1	1	1
10	2	2	2	2
11	3	3	3	3
100	4	4	4	4
101	5	5	5	5
110	6	6	6	6
111	7	7	7	7
1000	10	8	8	8
1001	11	9	9	9
1010	12	10	A	A
1011	13	11	B	B
1100	14	12	C	C
1101	15	13	D	D
1110	16	14	E	E
1111	17	15	F	F
10000	20	16	10	10

Při převodu do desítkové soustavy počítáme sumu "číslice krát základ na N-tou":

- Jednotky vždy mají mocninu na 0-tou a nalevo od nich se mocniny zvyšují.

Převod z dvojkové do desítkové:

$$(1101)_2 = 1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0 = 8 + 4 + 0 + 1 = (13)_{10}$$

Převod z osmičkové do desítkové:

$$(456)_8 = 4 * 8^2 + 5 * 8^1 + 6 * 8^0 = 256 + 40 + 6 = (302)_{10}$$

Převod z šestnáctkové do desítkové:

$$(123)_{16} = 1 * 16^2 + 2 * 16^1 + 3 * 16^0 = 256 + 32 + 3 = (291)_{10}$$

Vyzkoušejte si:

- Čísla převedte do desítkové soustavy.

$$(11\ 0010)_2 = (\dots)_{10}$$

$$(555)_8 = (\dots)_{10}$$

$$(234)_{16} = (\dots)_{10}$$

$$(1101\ 0110)_2 = (\dots)_{10}$$

$$(1\ 220)_{16} = (\dots)_{10}$$

Při převodu z desítkové soustavy **celočíslně dělíme** základem cílové soustavy, a výsledek získáme přečtením **zbytků po dělení** směrem **odspodu nahoru**:

Převod čísel z desítkové soustavy do:

dvojkové

$$\begin{aligned}(20)_{10} &= (1\ 0100)_2 \\ 20 / 2 &= 10 \text{ zb } 0 \\ 10 / 2 &= 5 \text{ zb } 0 \\ 5 / 2 &= 2 \text{ zb } 1 \\ 2 / 2 &= 1 \text{ zb } 0 \\ 1 / 2 &= 0 \text{ zb } 1\end{aligned}$$

osmičkové

$$\begin{aligned}(83)_{10} &= (123)_8 \\ 83 / 8 &= 10 \text{ zb } 3 \\ 10 / 8 &= 1 \text{ zb } 2 \\ 1 / 8 &= 0 \text{ zb } 1\end{aligned}$$

šestnáctkové

$$\begin{aligned}(140)_{10} &= (8C)_{16} \\ 140 / 16 &= 8 \text{ zb } 12 \\ 8 / 16 &= 0 \text{ zb } 8\end{aligned}$$

pozor

$$(12)_{10} = (C)_{16}$$

Vyzkoušejte si:

- Číslo převeďte z desítkové soustavy.

$$(180)_{10} = (\dots)_2$$

$$\dots / 2 = \dots \text{ zb } \dots$$

$$\dots / 2 = \dots \text{ zb } \dots$$

$$\dots / 2 = \dots \text{ zb } \dots$$

$$\dots / 2 = \dots \text{ zb } \dots$$

$$\dots / 2 = \dots \text{ zb } \dots$$

$$\dots / 2 = \dots \text{ zb } \dots$$

$$\dots / 2 = \dots \text{ zb } \dots$$

$$\dots / 2 = \dots \text{ zb } \dots$$

$$(105)_{10} = (\dots)_2$$

$$\dots / 2 = \dots \text{ zb } \dots$$

$$\dots / 2 = \dots \text{ zb } \dots$$

$$\dots / 2 = \dots \text{ zb } \dots$$

$$\dots / 2 = \dots \text{ zb } \dots$$

$$\dots / 2 = \dots \text{ zb } \dots$$

$$\dots / 2 = \dots \text{ zb } \dots$$

$$\dots / 2 = \dots \text{ zb } \dots$$

$$(93)_{10} = (\dots)_{16}$$

$$\dots / 16 = \dots \text{ zb } \dots$$

$$\dots / 16 = \dots \text{ zb } \dots$$

$$(200)_{10} = (\dots)_8$$

$$\dots / 8 = \dots \text{ zb } \dots$$

$$\dots / 8 = \dots \text{ zb } \dots$$

$$\dots / 8 = \dots \text{ zb } \dots$$

$$(357)_{10} = (\dots)_{16}$$

$$\dots / 16 = \dots \text{ zb } \dots$$

$$\dots / 16 = \dots \text{ zb } \dots$$

$$\dots / 16 = \dots \text{ zb } \dots$$

Pokud čísla převádíme mezi dvojkovou a šestnáctkovou (nebo osmičkovou) soustavou můžeme je převést bez počítání, rozdělením na **N-tice**:

- Jeden znak **šestnáctkové** odpovídá **čtyřem** znakům dvojkové soustavy ($16 = 2^4$).
- Jeden znak **osmičkové** odpovídá **třem** znakům dvojkové soustavy ($8 = 2^3$).

$$(3D0)_{16} = \left| \begin{array}{c|c|c} 3 & D & 0 \\ \hline 0011 & 1101 & 0000 \end{array} \right| = (11\ 1101\ 0000)_2$$

- Rozdělování vždy **začínáme z prava** a prázdná místa vlevo **doplňujeme nulami**:

$$(1\ 010\ 101\ 011)_2 = \left| \begin{array}{c|c|c|c} 001 & 010 & 101 & 011 \\ \hline 1 & 2 & 5 & 3 \end{array} \right| = (1253)_8$$

- Nejrychlejší způsob jak číslo převést z **šestnáctkové** do **osmičkové** (a obráceně) je přes **dvojkovou** soustavu:

$$(7B9)_{16} = \left| \begin{array}{c|c|c} 7 & B & 9 \\ \hline 0111 & 1011 & 1001 \\ \hline 011 & 110 & 111 & 001 \\ \hline 3 & 6 & 7 & 1 \end{array} \right| = (3671)_8$$

Vyzkoušejte si:

- Číslo převeďte bez desítkové soustavy.

$$(1234)_8 = \begin{array}{|c|c|c|c|} \hline \dots & \dots & \dots & \dots \\ \hline \dots & \dots & \dots & \dots \\ \hline \end{array} = (\dots)_2$$

$$(10111000101)_2 = \begin{array}{|c|c|c|c|} \hline \dots & \dots & \dots & \dots \\ \hline \dots & \dots & \dots & \dots \\ \hline \end{array} = (\dots)_8$$

$$(5CE6)_{16} = \begin{array}{|c|c|c|c|} \hline \dots & \dots & \dots & \dots \\ \hline \dots & \dots & \dots & \dots \\ \hline \end{array} = (\dots)_2$$

$$(11101110101001)_2 = \begin{array}{|c|c|c|c|} \hline \dots & \dots & \dots & \dots \\ \hline \dots & \dots & \dots & \dots \\ \hline \end{array} = (\dots)_{16}$$

$$(4537)_8 = \begin{array}{|c|c|c|c|} \hline \dots & \dots & \dots & \dots \\ \hline \dots & \dots & \dots & \dots \\ \hline \dots & \dots & \dots & \dots \\ \hline \dots & \dots & \dots & \dots \\ \hline \end{array} = (\dots)_{16}$$

Pevná řádová čárka

U **desetinných čísel** rozlišujeme mezi čísla s **pevnou** a **plovoucí** řádovou čárkou:

- Čísla s **pevnou** řádovou čárkou mají **pevně danou přesnost** a pracujeme s nimi stejně jako s celými čísly.
- Čísla s **plovoucí** řádovou čárkou (**float**) mají **proměnlivou přesnost** a práce s nimi vyžaduje **matematický koprocessor**.
- Přesnost obou typů čísel je omezená – jejich hodnoty se **ořezávají** (**truncation**).

Na cvičeních ISU:

- 1. až 10. týden budeme pracovat pouze s **celými čísly**.
- 11. až 13. týden budeme používat čísla s **plovoucí řádovou čárkou**.
- Čísla s **pevnou řádovou čárkou** proto na cvičeních přeskochíme.

Při převodu do desítkové soustavy počítáme sumu "číslice krát základ na N-tou":

- Jednotky vždy mají mocninu na 0-tou, nalevo od nich se mocniny zvyšují a napravo od nich se mocniny snižují.

Převod z dvojkové do desítkové:

$$\begin{aligned}(110.101)_2 &= 1 * 2^2 + 1 * 2^1 + 0 * 2^0 + 1 * 2^{-1} + 0 * 2^{-2} + 1 * 2^{-3} = \\&= 1 * 4 + 1 * 2 + 1 * \frac{1}{2} + 1 * \frac{1}{8} = \\&= 4 + 2 + 0.5 + 0.125 = (6.625)_{10}\end{aligned}$$

Převod z šestnáctkové do desítkové:

$$\begin{aligned}(AB.CD)_{16} &= A * 16^1 + B * 16^0 + C * 16^{-1} + D * 16^{-2} = \\&= 10 * 16 + 11 * 1 + 12 * \frac{1}{16} + 13 * \frac{1}{256} = \\&= 160 + 11 + 0.75 + 0.05078125 = (171.80078125)_{10}\end{aligned}$$

Při převodu z desítkové soustavy:

- Celou část **celočíslně dělíme** základem cílové soustavy a výsledek získáme přečtením **zbytků po dělení** směrem **odspodu nahoru**.
- Desetinnou část **násobíme** základem cílové soustavy a výsledek získáme přečtením **přenosů** přes desetinnou čárku směrem **shora dolů**.

Převod desetinného čísla z desítkové do **dvojkové** soustavy:

$$(12.8125)_{10} = (12)_{10} + (0.8125)_{10} = (1100)_2 + (0.1101)_2 = (1100.1101)_2$$

$$\begin{array}{l} (12)_{10} = (1100)_2 \\ 12 / 2 = 6 \text{ zb } 0 \\ 6 / 2 = 3 \text{ zb } 0 \\ 3 / 2 = 1 \text{ zb } 1 \\ 1 / 2 = 0 \text{ zb } 1 \end{array}$$

$$\begin{array}{l} (0.8125)_{10} = (0.1101)_2 \\ 0.8125 * 2 = 1 .625 \\ 0.625 * 2 = 1 .25 \\ 0.25 * 2 = 0 .5 \\ 0.5 * 2 = 1 .0 \end{array}$$

Pokud čísla převádíme mezi dvojkovou a šestnáctkovou (nebo osmičkovou) soustavou můžeme je převést i bez počítání, rozdělením na **N-tice**:

- Jeden znak **šestnáctkové** odpovídá **čtyřem** znakům dvojkové soustavy.
- Jeden znak **osmičkové** odpovídá **třem** znakům dvojkové soustavy.
- Rozdělování vždy začínáme na **pozici desetinné čárky** a případná prázdná místa z obou stran **doplňujeme nulami**:

$$(34.56)_8 = \left| \begin{array}{c|c|c|c} 3 & 4 & 5 & 6 \\ \hline 011 & 100 & 101 & 110 \end{array} \right| = (11\ 100.101\ 11)_2$$

$$(AB.CD)_{16} = \left| \begin{array}{c|c|c|c} A & B & C & D \\ \hline 1010 & 1011 & 1100 & 1101 \end{array} \right| = (1010\ 1011.1100\ 1101)_2$$

$$(11\ 100\ 110.001\ 101)_2 = \left| \begin{array}{c|c|c|c|c|c} 011 & 100 & 110 & 001 & 101 \\ \hline 3 & 4 & 6 & 1 & 5 \end{array} \right| = (346.15)_8$$

$$(10\ 0110.1011\ 111)_2 = \left| \begin{array}{c|c|c|c} 0010 & 0110 & 1011 & 1110 \\ \hline 2 & 6 & B & E \end{array} \right| = (26.BE)_{16}$$

Doplňkový kód

- V ISU budeme používat procesory s architekturou IA-32 (rodina x86).
- Tyto procesory celá čísla ukládají do registrů o délce 8, 16, nebo 32 bitů.
- Bity mohou obsahovat pouze nulu nebo jedničku (dvojková soustava).

EAX = AL =

AX =

- Čísla začínající **jedničkou** jsou **záporná**, čísla začínající **nulou** jsou **ne-záporná**.
- To jestli registr začínající jedničkou obsahuje **záporné číslo** nebo velké kladné **číslo bez znaménka** záleží jen na interpretaci dat – v **8b** registru platí **255 == -1**.

Doplňkový kód mění rozsah hodnot které lze v registru reprezentovat:

- Pro 8b registry z $\langle 0, 255 \rangle$ na $\langle -128, 127 \rangle$.
- Pro 16b registry z $\langle 0, 65535 \rangle$ na $\langle -32768, 32767 \rangle$.
- Pro 32b registry z $\langle 0, 4294967295 \rangle$ na $\langle -2147483648, 2147483647 \rangle$.

Znaménko čísla obrátíme operací **dvojkového doplněku** (**aritmetická negace**), která znamená **“invertuj všechny bity a přičti jedničku”**:

$$\begin{array}{r}
 (13)_{10} = 0\ 0\ 0\ 0\ 1\ 1\ 0\ 1 \\
 \hline
 \text{inv}(13) = 1\ 1\ 1\ 1\ 0\ 0\ 1\ 0 \\
 + 1 = 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1 \\
 \hline
 = 1\ 1\ 1\ 1\ 0\ 0\ 1\ 1 = (-13)_{10}
 \end{array}$$

$$\begin{array}{r}
 (-13)_{10} = 1\ 1\ 1\ 1\ 0\ 0\ 1\ 1 \\
 \hline
 \text{inv}(-13) = 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0 \\
 + 1 = 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1 \\
 \hline
 = 0\ 0\ 0\ 0\ 1\ 1\ 0\ 1 = (13)_{10}
 \end{array}$$

Vyzkoušejte si:

- Čísla v **desítkové** soustavě převedte do **doplňkového kódu** na osmi bitech.

$$35 = \dots$$

$$-23 = \text{inv}(23) + 1 = \text{inv}(\dots) + 0000\ 0001 = \dots$$

$$-97 = \text{inv}(97) + 1 = \text{inv}(\dots) + 0000\ 0001 = \dots$$

- Čísla v **doplňkovém kódu** na osmi bitech převedte do **desítkové** soustavy.

$$0011\ 0110 = \dots$$

$$1101\ 1001 = -(\text{inv}(1101\ 1001) + 0000\ 0001) = -(\dots) = \dots$$

$$1000\ 1000 = -(\text{inv}(1000\ 1000) + 0000\ 0001) = -(\dots) = \dots$$

S hodnotami uloženými v registrech můžeme provádět **aritmetické operace**:

- Při sčítání několika jedniček dochází k **přenosu** (carry) do vyššího řádu.

0	0	1	1	0	0	1	1	sčítanec
+0	+1	+0	+1	+0	+1	+0	+1	sčítanec
				+1	+1	+1	+1	přenos
0	1	1	10	1	10	10	11	součet

$$\begin{array}{r}
 108 = 01101100 \\
 + 42 = 00101010 \\
 \hline
 111 \\
 10010110 = 150
 \end{array}$$

přenosy

Protože registry mají omezenou velikost, výsledek výpočtu se do nich **nemusí** vejít:

- Při počítání **bez znaménka** chyba nastává jen při přenosu **Z** nejvyššího bitu:

$$\begin{array}{r}
 100 = 01100100 \\
 + 200 = 11001000 \\
 \hline
 11 \\
 00101100 = 44 \text{ (chyba)}
 \end{array}$$

přenosy

Doplňkový kód umožňuje aritmetické operace provádět i s čísly **se znaménkem**:

- Při počítání **se znaménkem** je chyba při přenosu jen **DO** nebo **Z** nejvyššího bitu.
- Chybě při počítání se znaménkem se říká **přetečení** (overflow).

$$\begin{array}{r|l}
 -108 = 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0 \\
 + -42 = 1\ 1\ 0\ 1\ 0\ 1\ 1\ 0 \\
 \hline
 1\quad\quad 1\quad 1 & \text{přenos jen } \mathbf{Z} \\
 0\ 1\ 1\ 0\ 1\ 0\ 1\ 0 & = 106\ (\text{chyba})
 \end{array}$$

$$\begin{array}{r|l}
 108 = 0\ 1\ 1\ 0\ 1\ 1\ 0\ 0 \\
 + 42 = 0\ 0\ 1\ 0\ 1\ 0\ 1\ 0 \\
 \hline
 1\ 1\quad 1 & \text{přenos jen } \mathbf{DO} \\
 1\ 0\ 0\ 1\ 0\ 1\ 1\ 0 & = -106\ (\text{chyba})
 \end{array}$$

- Pokud při počítání **se znaménkem** dojde k přenosu **DO i Z** nejvyššího bitu, tak se spolu obě chyby vzájemně vyruší a výsledek **bude platný**!

$$\begin{array}{r|l}
 100 = 0\ 1\ 1\ 0\ 0\ 1\ 0\ 0 \\
 + -56 = 1\ 1\ 0\ 0\ 1\ 0\ 0\ 0 \\
 \hline
 1\ 1 & \text{přenos } \mathbf{DO\ i\ Z} \\
 0\ 0\ 1\ 0\ 1\ 1\ 0\ 0 & = 44\ (\text{ok})
 \end{array}$$

Vyzkoušejte si:

- Sečtěte čísla v **doplňkovém kódu** na osmi bitech, a rozhodněte jestli je výsledek platný nebo jestli při výpočtu došlo k **přetečení**.

$$\begin{array}{r|l}
 49 = & 0\ 0\ 1\ 1\ 0\ 0\ 0\ 1 \\
 +\ -99 = & 1\ 0\ 0\ 1\ 1\ 1\ 0\ 1 \\
 \hline
 & \begin{array}{l} \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot \\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot \end{array} \begin{array}{l} \text{přenosy} \\ = \dots \end{array}
 \end{array}$$

$$\begin{array}{r|l}
 -123 = & 1\ 0\ 0\ 0\ 0\ 1\ 0\ 1 \\
 +\ -45 = & 1\ 1\ 0\ 1\ 0\ 0\ 1\ 1 \\
 \hline
 & \begin{array}{l} \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot \\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot \end{array} \begin{array}{l} \text{přenosy} \\ = \dots \end{array}
 \end{array}$$

$$\begin{array}{r|l}
 -70 = & 1\ 0\ 1\ 1\ 1\ 0\ 1\ 0 \\
 +\ -58 = & 1\ 1\ 0\ 0\ 0\ 1\ 1\ 0 \\
 \hline
 & \begin{array}{l} \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot \\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot \end{array} \begin{array}{l} \text{přenosy} \\ = \dots \end{array}
 \end{array}$$

$$\begin{array}{r|l}
 70 = & 0\ 1\ 0\ 0\ 0\ 1\ 1\ 0 \\
 +\ 58 = & 0\ 0\ 1\ 1\ 1\ 0\ 1\ 0 \\
 \hline
 & \begin{array}{l} \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot \\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot \end{array} \begin{array}{l} \text{přenosy} \\ = \dots \end{array}
 \end{array}$$