Čísla, číselné soustavy a kódování

Michal Šerý

Technicné principy počítačů

Obsah

Pojmy

Číslo

Vyjadřuje množství něčeho.

Číslice

Symbol.

Číselná soustava

Číselná soustava je způsob reprezentace čísel.

- Unární
- Nepoziční
- Poziční

Pojmy

Základ soustavy, báze (ang. radix)

Značí se r nebo z, a je to obvykle kladné celé číslo definující maximální počet číslic, které jsou v dané soustavě k dispozici.

Unární soustava

Příklad

jedničková (unární, r=1) – přestože si to ani neuvědomujeme, tuto soustavu běžně používáme při počítání na prstech nebo při psaní čárek označujících počet piv na účet v restauračních zařízeních. Může být řazena mezi speciální poziční soustavy nebo i zcela mimo dělení na poziční/nepoziční soustavy.

Nepoziční soustava

Příklady

- Římské číslice
- Egyptské číslice
- Řecké číslice
- Etruské číslice
- Unární soustava (může být i poziční)

Nepoziční soustava

Římské číslice

Římské číslo	Arabské číslo
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

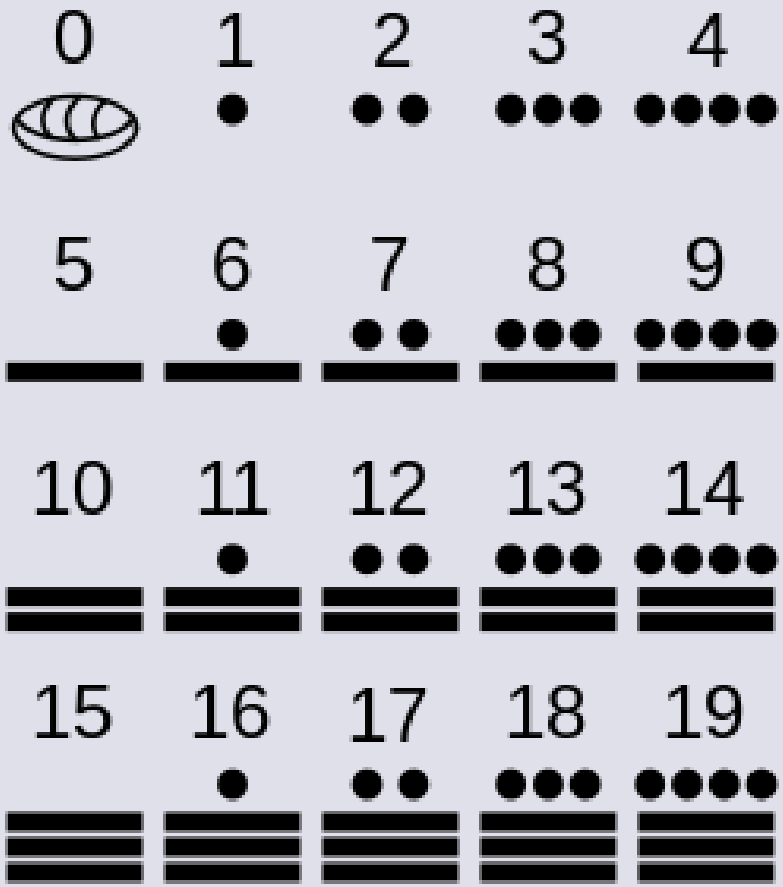
MMXVI = 2016

Poziční soustava - polyadické číselné soustavy

Mezi nejčastěji používané poziční číselné soustavy patří:

- desítková (decimální, dekadická, r=10) – nejpoužívanější v běžném životě
- dvanáctková (r=12) – dnes málo používaná, ale dodnes z ní zbyly názvy prvních dvou řádů – tucet a veletucet
- šedesátková (r=60) – používá se k měření času pro zlomky hodiny; číslice se obvykle zapisují desítkovou soustavou jako 00 až 59 a řády se oddělují dvojtečkou; staré názvy prvních dvou řádů jsou kopa a velekopa.

Příklad dvacítková (r=20) – mayové



Obrázek: Mayské číslice

(zdroj Wikipedie)

Věta (O reprezentaci přirozených čísel (včetně 0))

Libovolné přirozené číslo N (včetně 0) lze vyjádřit jako součet mocninné řady o základu $r \geq 2; r \in \mathbb{N}$:
Číslo N se (v poziční číselné soustavě o základu r) zapisuje jako řetěz číslic (symbolů) S_i pro koefcienty a_i zleva v pořadí pro i od $n-1$ k 0:
$(S_{n-1} S_{n-2} \ldots S_1 S_0)_r$

- dvojková (binární, $r = 2$) – číslice **0**, **1** přímá implementace v digitálních elektronických obvodech (použitím logických členů - moderní počítače)
- osmičková (oktální, oktalová, $r = 8$) – číslice **0** až **7**
- šestnáctková (hexadecimální, $r = 16$) – číslice **0** až **9** a **A** až **F**

Poziční soustava

$$(1024,48)_{10}$$

Váhový polynom

Libovolné číslo N (včetně 0) lze vyjádřit jako součet mocninné řady o základu $r \geq 2; r \in \mathbb{N}$:
Poziční systém s jedním nebo více základy

$$N = \sum_{i=-m}^{n} S_i z_i$$

- N - číslo
- S_i - řádová číslice
- z_i - základ řádu - váha

Základ řádu - váha - může být stanovena podle následujícího vztahu.

$$z_i = r^i$$

Ukázka - celé číslo

$$N = a_{n-1} \cdot z^{n-1} + a_{n-2} \cdot z^{n-2} + \ldots + a_1 \cdot z^1 + a_0$$

nebo

$$N = (\ldots((a_{n-1} \cdot z + a_{n-2}) \cdot z + a_{n-3}) \cdot z + \ldots + a_1) \cdot z + a_0$$

Ukázka - desetinné číslo

$$D = d_{-1} \cdot z^{-1} + d_{-2} \cdot z^{-2} + \ldots + d_{-m+1} \cdot z^{-m+1} + d_{-m} \cdot z^{-m}$$

nebo

$$D =$$

$$(\ldots((d_{-m} \cdot z^{-1} + d_{-m+1}) \cdot z^{-1} + d_{-m+2}) \cdot z^{-1} + \ldots + d_{-2}) \cdot z^{-1} + d_{-1}) \cdot z^{-1}$$

Ukázka

$$(1024,48)_{10} = 1 \cdot 10^3 + 0 \cdot 10^2 + 2 \cdot 10^1 + 4 \cdot 10^0 + 4 \cdot 10^{-1} + 8 \cdot 10^{-2}$$

$$(1011,01)_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2}$$

Pokud to vyčíslíme:

$$(1011,01)_2 = 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 + 0 \cdot 0,5 + 1 \cdot 0,25$$

$$(1011,01)_2 = 8 + 2 + 1 + 0,25 = (11,25)_{10}$$

Mocniny dvou

Tabulka

i	velikost	i	velikost
2^0	1	2^{10}	1 024
2^1	2	2^{11}	2 048
2^2	4	2^{12}	4 096
2^3	8	2^{13}	8 024
2^4	16	2^{14}	16 384
2^5	32	2^{15}	32 768
2^6	64	2^{16}	65 536
2^7	128	2^{17}	131 072
2^8	256	2^{18}	262 144
2^9	512	2^{19}	524 288

Mocniny dvou

Tabulka

i	velikost	předpona
2^{10}	1 024	K
2^{20}	1 048 576	M
2^{30}	1 073 741 824	G
2^{40}	1 099 511 627 776	T

Tabulka

desítkově	binárně	oktalově	hexadecimálně
0	0000	00	0
1	0001	01	1
2	0010	02	2
3	0011	03	3
4	0100	04	4
5	0101	05	5
6	0110	06	6
7	0111	07	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Převod 10 -> 2 (potažmo 10 -> libovolné)

Postup

Rozdělíme si číslo na celou a desetinnou část.
Celou část vezmeme a začneme celočíselně dělit 2.
Sepisujeme zbytky.
Zbytky napíšeme odzadu.
Desetinnou část vezmeme a začneme „celočíselně“ násobit 2.
Sepisujeme celé části.

$$725,04$$

Postup - celá část

$$725 : 2 = 362 \quad (1)$$

$$362 : 2 = 181 \quad (0)$$

$$181 : 2 = 90 \quad (1)$$

$$90 : 2 = 45 \quad (0)$$

$$45 : 2 = 22 \quad (1)$$

$$22 : 2 = 11 \quad (0)$$

$$11 : 2 = 5 \quad (1)$$

$$5 : 2 = 2 \quad (1)$$

$$2 : 2 = 1 \quad (0)$$

$$1 : 2 = 0 \quad (1)$$

Postup - desetinná část

$$0,04 \cdot 2 = 0,08 \quad (0)$$

$$0,08 \cdot 2 = 0,16 \quad (0)$$

$$0,16 \cdot 2 = 0,32 \quad (0)$$

$$0,32 \cdot 2 = 0,64 \quad (0)$$

$$0,64 \cdot 2 = 0,28 \quad (1)$$

$$0,28 \cdot 2 = 0,56 \quad (0)$$

$$0,56 \cdot 2 = 0,12 \quad (1)$$

$$0,12 \cdot 2 = 0,24 \quad (0)$$

$$0,24 \cdot 2 = 0,48 \quad (0)$$

$$0,48 \cdot 2 = 0,96 \quad (0)$$

$$0,96 \cdot 2 = 0,92 \quad (1)$$

Převod 10 -> 2

Výsledek

$$(725,04)_{10} \doteq (1011010101,00001010001\cdots)_2$$

Převody

Zvláštní případ:

Použití pozičních číselných soustav o základu $2^k (k \in \mathbb{N})$:

- dvojková (binární, r=2)
- osmičková (oktální, oktalová, r=8)
- šestnáctková (hexadecimální, r=16)

Použití pozičních číselných soustav o základu $2^k (k \in N)$:

Převod zápisu čísla v soustavě o základu $r^k (k \in N)$ na zápis v soustavě o základu r (a naopak):

Každá číslice z čísla soustavy o základu r^k nahradíme k-ticí symbolů soustavy o základu r.
A naopak, k-tice symbolů v zápisu, brány od řádové čárky, (chybějící symboly nahrazeny 0)

Soustavy

dvojková	1001110110110 10,01101
šestnácková	4 E D A , 6 8
dvojková	100111011011010,01101
osmičková	4 7 3 3 2 , 3 2

Obrázek: Soustavy 2<->16; 2<->8

Příklady na procvičení

10 -> 2

152
483
128
444,4
32,5
14,75
5,125
6,875

Příklady na procvičení - řešení

10 -> 2

$$(152)_{10} = (10011000)_2$$

$$(483)_{10} = (111100011)_2$$

$$(128)_{10} = (10000000)_2$$

$$(444,4)_{10} = (110111100,\overline{0110})_2$$

$$(32,5)_{10} = (100000,1)_2$$

$$(14,75)_{10} = (1110,11)_2$$

$$(5,125)_{10} = (101,001)_2$$

$$(6,875)_{10} = (110,111)_2$$

Příklady na procvičení - řešení

Pozor

Při převodu desetinných čísel do binární soustavy jen ve výjimečných případech získate číslo s konečným počtem desetinných míst! Většinou převod končí s určitou chybou výsledku při konečném počtu míst.

Některé možnosti

- Čísla v kódu NBCD (BCD) - číslice
- Celá čísla bez znaménka - unsigned integer
- Celá čísla se znaménkem - signed integer
- Čísla v pohyblivé řádové čárce - float (IEEE 754)

$$c = M \cdot z^E$$

c . . . číslo
M . . . mantisa
z . . . základ soustavy
E . . . exponent

Kódování záporných čísel

- přímý kód
- inverzní kód
- doplňkový kód
- aditivní kód

Kódování záporných čísel

Přímý kód

- nejvyšší bit (první zleva) má význam znaménka
- 0 odpovídá kladnému číslu
- 1 odpovídá zápornému číslu
- máme dvě nuly (+0 a -0) - problém

$$(45)_{10} = (00101101)_2$$

$$(-45)_{10} = (10101101)_2$$

Kódování záporných čísel

Inverzní kód

- kladná normálně
- záporná - bitová negace
- máme dvě nuly (+0 a -0) - problém

$$(45)_{10} = (00101101)_2$$

$$(-45)_{10} = (11010010)_2$$

Kódování záporných čísel

Doplňkový kód

- kladná normálně
- záporná - až do první jedničky odzadu opíšeme a pak bitově negujeme
- máme jednu nulu

$$(46)_{10} = (00101110)_2$$

$$(-46)_{10} = (11010010)_2$$

Tento kód se používá v:
kódování celých čísel (integer, long, ...).
kódování mantisy v reprezentaci čísel v pohyblivé řádové čárce (single, double, ...).

Kódování záporných čísel

Aditivní kód (s posunutou nulou)

Např.: na 8 bitech. Rozsah $0 \div 255$ potom
$(-128)_{10}$ odpovídá $(0000\ 0000)_2$
$(0)_{10}$ odpovídá $(1000\ 0000)_2$
$(127)_{10}$ odpovídá $(1111\ 1111)_2$

Tento kód se používá v kódování exponentu při reprezentaci čísel v pohyblivé řádové čárce.

ASCII tabulka

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

Unicode

Unicode (anglicky Unicode) je technická norma pro oblast výpočetní techniky definující jednotnou znakovou sadu a konzistentní kódování znaků pro reprezentaci a zpracovávání textů použitelné pro většinu písem používaných v současnosti na Zemi. Unicode je vyvíjen v součinnosti s ISO/IEC 10646,
Nejnovější verze obsahuje repertoár více než 140 000 znaků pokrývajících 154 moderních a historických písem a mnoho sad symbolů. Standard sestává ze sady tabulek pro vizuální referenci, popisu metod kódování, atd.
Poslední verze je Unicode 13.0 z března roku 2020 (143859 znaků ve 154 písmech).
Normu udržuje Unicode Consortium.

`https://cs.wikipedia.org/wiki/Unicode`

Unicode

Původně bylo kódování Unicode navrhováno jako 16bitové (65536 možností). Posléze bylo rozšířeno (kvůli čínským znakům) rozšířeno na 32bitové (více jak 4 miliardy).

`https://cs.wikipedia.org/wiki/Unicode`

Principy standardu Unicode

- **Jednotnost** – konstantní šířka znaků (UTF-32) dovoluje efektivní hledání, třídění, editaci a zobrazení prvků.
- **Univerzálnost** – zahrnutí všech znaků, které by mohly být využity při výměně textů – především ty, které už byly definovány v hlavních mezinárodních, národních a průmyslových znakových sadách.
- **Jednoznačnost** – jakákoli 16bitová (dnes 32bitová) hodnota zastupuje v jakémkoliv kontextu stejný znak.
- **Maximální využití** – snadná zpracovatelnost textu poskládaného z posloupnosti znaků o konstantní šířce; kódování znaků není závislé na kontextu, pro strojové zpracování textu není nutné vyhodnocovat escape sekvence nebo prohledávat text dopředu či zpět kvůli určení totožnosti znaků.

https://cs.wikipedia.org/wiki/Unicode

Některé zkratky spojené s Unicode

- **UCS** - Universal Character Set.
- **BMP** - Basic Multilingual Plane – základní vícejazyčná rovina Unicode. Původní rozsah Unicode, tj. prvních 65 536 znaků.
- **BOM** - Byte Order Mark - speciální značky umístěné na začátku textu.

`https://cs.wikipedia.org/wiki/Unicode`

Endianita

Pořadí bajtů, anglicky byte order, je v informatice způsob uložení čísel (kódů) v operační paměti počítače, který definuje, v jakém pořadí se uloží jednotlivé bajty datového typu. Jde o datové typy, které zabírají více než jeden bajt.

Etymologie

Přídavné jméno endian má původ ve spisech anglo-irského spisovatele 18. století Jonathana Swifta . V románu Gulliverovy cesty z roku 1726 vykresluje konflikt mezi sektami Lilliputianů rozdělenými na ty, které rozbíjejí skořápku vařeného vejce z velkého konce nebo z malého konce. Nazval je „Big-Endians“ a „Little-Endians“. Danny Cohen zavedl pojmy big-endian a little-endian do počítačové vědy pro objednávání dat v **Internet Experiment Note** publikované v roce 1980.

`https://cs.qaz.wiki/wiki/Endianness`.

Little-endian

Na paměťové místo s nejnižší adresou uloží nejméně významný bajt (LSB) a za něj se ukládají ostatní bajty až po nejvíce významný bajt (MSB). (Mnemotechnická pomůcka: little end first) a patří mezi ně MOS Technology 6502, Intel x86 a DEC VAX.

Big-endian

Na paměťové místo s nejnižší adresou uloží nejvíce významný bajt (MSB) a za něj se ukládají ostatní bajty až po nejméně významný bajt (LSB) na konci. (Mnemotechnická pomůcka: big end first) a patří mezi ně Motorola 68000, SPARC a System/370.

Middle-endian (nebo někdy mixed-endian)

Některé architektury užívají složitější způsob pro určení pořadí jednotlivých bajtů, který je dán kombinací obou výše zmíněných způsobů. Mezi takovéto architektury patří např. rodina procesorů PDP-11. Tento formát je také použit pro ukládání čísel s pohyblivou

Základní kódování Unicode

- **UTF-8** - kóduje znaky různě dlouhou posloupností bajtů podle jejich kódu v Unicode (1–4 bajty, pro původní 31bitové ISO/IEC 10646 až 6 bajtů).
- **UTF-16** - znaky BMP reprezentují jedním 16bitovým číslem, znaky mimo BMP jsou reprezentovány párem 16bitových čísel. UTF-16BE (big-endian), UTF-16LE (little-endian) a UTF-16 (nestanoveno, může být určeno pomocí BOM).
- **UTF-32** - též označováno jako UCS-4, každý znak reprezentován přímo 32bitovým číslem. UTF-32BE (big-endian), UTF-32LE (little-endian) a UTF-32 (nestanoveno, může být určeno pomocí BOM).

`https://cs.wikipedia.org/wiki/Unicode`

Kódování češtiny - jeden z problémů

Pro český jazyk se používalo nejméně 5 různých kódování

- kódování bratří Kamenických
- PC Latin 2
- Windows-1250
- ISO Latin 2
- KOI8-CS

V dnešní době je tento problém víceméně odstraněn, ale stále je možné se s ním v některých speciálních případech setkat.

Příklady kódování

Samuel F. B. Morse (27.4.1791-2.4.1872)

Již v září 1837, byl předveden veřejnosti nový Morseův telegrafní přístroj využívající nový způsob kódování znaků. A tak amatérský fyzik a malíř Samuel F. B. Morse patentoval svůj vynález a získal tím i vypsanou cenu Kongresu. Jeho způsob kódování využíval sériový přenos dat, což jak se ukázalo, bylo v oné době, kdy funkci kodéru a dekodéru zastával člověk, zjevně to nejlepší. Ještě za života se pan Samuel F. B. Morse dočkal několika realizací svého vynálezu. Ta první byla v květnu roku 1844 na trase, jejíž délka byla 60 km, a to mezi městy Washington a Baltimore.

Morseův princip

Spočívá v tom, že písmenům s velkou pravděpodobností výskytu jsou přiřazeny kódová slova s kratšími časovými intervaly a naopak.

Zdroj: Kódování

Čárové kódy

Typy

- CODE39 : 1. v roce 1974
- Industrial 2/5
- UPC a EAN : Obchod
- INTERLAVED : Kontejnerová přeprava
- PDF417 : 2D
- QR
- ...

Zdroj: Kódování

Industrial 2/5

Jedná se o čistě numerický kód proměnné délky. Kód je tvořen znakem start, příslušným počtem datových znaků (číslice 0 až 9) a znakem stop. Kód každého znaku je tvořen 5 čarami, z nichž jsou 3 úzké a 2 široké. Poměr šířky široké a úzké čáry je 3:1. Mezery nenesou žádnou informaci a slouží jen k oddělení čar.

Zdroj: Kódování

Industrial 2/5

Znak	1. čára	2. čára	3. čára	4. čára	5. čára
0	0	0	1	1	0
1	1	0	0	0	1
2	0	1	0	0	1
3	1	1	0	0	0
4	0	0	1	0	1
5	1	0	1	0	0
6	0	1	1	0	0
7	0	0	0	1	1
8	1	0	0	1	0
9	0	1	0	1	0
start	1	1	0	N/A	N/A
stop	1	0	1	N/A	N/A

Obrázek: Ukázka

Zdroj: Kódování

Kódování znaků

Co dodělat:

- příklady výpočtu v různých kódováních (sčítání/odčítání)