

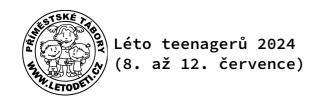
2.1 Binární soustava

Číselné soustavy používáme pro reprezentaci nějakého počtu, množství něčeho, jednoduše pro reprezentaci čísel. Znáš už desítkovou soustavu, bereme ji jako základní soustavu pro počítání čehokoli, počítače to mají ale jinak, používají pro počítání a ukládání dat soustavu binární neboli dvojkovou. Desítková používá 10 číslic (0 až 9), binární ale pouze 2 (0 a 1). Pro nás informatiky je důležité rozumět převodům mezi desítkovou a binární soustavou, jelikož pokud máme něco uložené v paměti počítače, tak to jsou pro nás pouze jedničky a nuly, potřebujeme si to tedy převést do desítkové soustavy aby nám to dávalo smysl (popřípadě přeložíme uložené jedničky a nuly podle nějakého kódování na písmena, o tom ale později).

Jedna jednička nebo nula se nazývá **bit.** V paměti jsou informace uložené po blocích, typicky po 8 bitech a tomuto bloku říkáme **bajt**. Tedy jeden bajt může vypadat například: 1010 1101 nebo 1101 0100. Všimni si že pro lepší čitelnost píšeme tyhle bloky **s mezerou uprostřed**.

Proč používají počítače binární soustavu?

Jak	si	počítače	mezi	sebou	posílají	informace	(data)?



Každý jeden bit v čísle zapsaném v binární soustavě má svou **hodnotu** v **desítkové soustavě**, díky tomu můžeme celkem jednoduše mezi těmito soustavami **převádět**, prostě jen sečteme **hodnoty bitů na pozicích kde je jednička** (z binární do desítkové) nebo **umístíme jedničky** na pozice binárního čísla tak, aby nám dávalo nějaké v desítkové soustavě (z desítkové do binární).

Tabulka hodnot pozic:

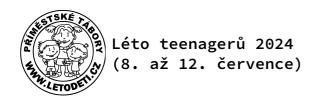
Číslo pozice	8.	7.	6.	5.	4.	3.	2.	1.
Hodnota pozice v desítkové soustavě	128	64	32	16	8	4	2	1
Mocnina dvojky	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2º
Číslo v binární soustavě	1	1	1	1	1	1	1	1

Příklady převodů z binární do desítkové:

Číslo v binární soustavě	Výpočet převodu podle hodnot pozic	Číslo v desítkové soustavě
0000 0011	2+1	3
0010 0010	32+2	32
0110 0100	64+32+4	100
1001 0001	128+16+1	145
0011 1101	32+16+8+4+1	61

Příklady na procvičení převodů z binární do desítkové:

Číslo v binární soustavě	Výpočet převodu podle hodnot pozic	Číslo v desítkové soustavě
0000 0100		
0001 0001		
0000 1100		
0001 0110		
0010 1000		



Příklady převodů z desítkové do binární:

Číslo v desítkové soustavě	Výpočet převodu podle hodnot pozic	Číslo v binární soustavě
15	8+4+2+1	0000 1111
23	16+4+2+1	0001 0110
48	32+16	0011 0000
132	128+4	1000 0100
0	0	0000 0000

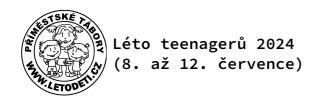
Příklady na procvičení převodů z desítkové do binární:

Číslo v desítkové soustavě	Výpočet převodu podle hodnot pozic	Číslo v binární soustavě
8		
12		
17		
23		
41		

2.2 Počítání v binární soustavě

Počítače počítají v binární soustavě, nás bude zajímat hlavně sčítání, jak to ale dělají? Funguje to podobně jako sčítání pod sebou v desítkové soustavě (což znáte asi tak ze třetí třídy). Sčítáme tedy postupně jednotlivé cifry a speciálně pokud tedy sčítáme dvě jedničky dostaneme nulu a jedna jde dál, sčítání nuly a jedničky je jednička a sčítání tří jedniček (jedna z předchozího sčítání), pak je jednička a jedna jde dál. Lépe to lze ale pochopit na příkladech níže (můžete si pak výsledek i sčítance převést do desítkové soustavy a sečtené čísla v desítkové soustavě a výsledek by mělo být stejné číslo, zkuste to pro ověření).

Informatika trochu jinak



Příklady:

Příklady k procvičení sčítání:

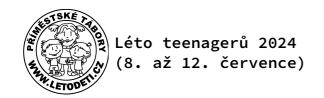
Zatím jsme pracovali jen s čísly kladnými a nulou (tedy nezápornými), v počítači často chceme reprezentovat i **čísla záporná** (např. -3, -12, ...), to děláme pomocí tzv. **dvojkového doplňku**. Ten vypadá tak, že kladnému číslu k číslu které chceme mít záporné (tedy číslo opačné) nejdříve **invertujeme** všechny bity (jinak i negujeme nebo obrátíme) tak, že **z jedničky uděláme nulu a opačně** a následně k výsledku **přičteme** jednu **jedničku**. Důležité je vědět, že díky dvojkovému doplňku pořád **funguje sčítání** čísel (tedy můžeme sečíst např. 8 a -3) a tedy umíme už i **odečítat**.

Příklady převodu do dvojkového doplňku (na záporné číslo):

Záporné číslo v desítkové soustavě	Opačné číslo k tomuto číslu v binární soustavě	Invertované číslo v binární soustavě	Číslo s přičtenou jedničkou (už dvojkový doplněk)
-4	0000 0100	1111 1011	1111 1100
-7	0000 0111	1111 1000	1111 1001

Příklady k procvičení převodu do dvojkového doplňku:

Záporné číslo v desítkové soustavě	Opačné číslo k tomuto číslu v binární soustavě	v binární	Číslo s přičtenou jedničkou
-2			
-10			

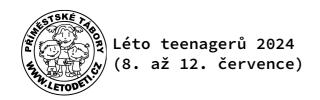


Informatika trochu jinak

Teď si to vše **spojíme dohromady**, čísla v závorkách nejdříve **převeďte** do binární soustavy a následně **spočtěte** v binární soustavě příklad, výsledek převeďte do desítkové soustavy a zkontrolujte, že vám to vyšlo správně:

- 1) (1+4)
- 2) (4+5) 3) (7+12) 4) (8-1)
- 5) (12-4)

Prostor pro mezivýpočty a poznámky:



2.3 Kódování písmen v počítači

Kódování (nebo jinak i reprezentace) písmen a tedy i textu v počítači je vyřešeno překladovou tabulkou, která obsahuje číselný kód pro písmena (a, b, c, ...) a speciální znaky (@, 8, !, ...). Tedy každé číslo v této tabulce znamená nějaké písmeno či znak. Slova a věty jsou tedy representována čísly (s kterými už umíme pracovat a hlavně je převádět z desítkové do binární soustavy). Historicky vzniklo několik takovýchto překladových tabulek, nejúspěšnější a nejpoužívanější je ale ASCII tabulka a její následná rozšíření (např. UTF-8). My budeme pracovat se základním kódováním ASCII tabulky, kde jeden znak je 8 bitů, tedy jeden bajt, ta má ale svá omezení (např. neobsahuje písmena s háčky ani čárkami).

DLE SPC NUL SOH DC1 Q Α а q DC2 STX ETX DC3 C S S EOT DC4 \$ Т ENQ NAK % Ε U SYN ACK BEL ETB G g CAN BS Н Χ HT EM Ι Υ i) У SUB LF J j Ζ VTESC Κ { FF FS l < L GS CR = М } RS S0 > Ν US SI DEL

ASCII Tabulka

Tedy například slovo "Ahoj" je v počítači zapsáno jako čísla 81, 104, 111 a 106 (samozřejmě v binární soustavě).

Informatika trochu jinak

2.4 Šifrovačka s ASCII tabulkou

Vaším úkolem je vyluštit zakódované slova, jména s slovní spojení, později si o nich popovídáme. Mezivýpočty si pište kdekoli kde je volné místo.

1)	01001000	01100101	01101100	01101100	01101111
	•••••	•••••	•••••	•••••	•••••
	01010111	01101111	01110010	01101100	01100100
	•••••	•••••	•••••	•••••	•••••

2)	01001100	01101001	01101110	01110101	01110011			
	•••••	•••••	•••••	•••••	•••••			
	01010100	01101111	01110010	01110110	01100001	01101100	01100100	01110011
	•••••	•••••	•••••	•••••	•••••	•••••	•••••	•••••

3)	01010011	01110100	01100101	01110110	01100101
	•••••	•••••	•••••	•••••	•••••
	01001010	01101111	01100010	01110011	
	•••••	•••••	•••••	•••••	

4)	01001100	01101001	01101110	01110101	01111000
	•••••	•••••	•••••	•••••	•••••