

## 03 AKO KÓDUJE POČÍTAČ - TEXT

<i>Tematický celok / Téma</i>	<i>Stupeň školy / Odporúčaný ročník / Rozsah</i>
Reprezentácie a nástroje - informácie	SŠ / 1. ročník / 1 vyučovací hodina
<b>Požiadavky na vstupné vedomosti a zručnosti</b>	
<ul style="list-style-type: none"> <li>Informácia, kódovanie informácie</li> <li>Prevod čísel do postupnosti bitov (dvojkovej sústavy) a späť</li> </ul>	
<b>Ciele</b>	
<i>Žiakom osvojované vedomosti a zručnosti</i>	<i>Žiakom rozvíjané spôsobilosti</i>
<b>Reprezentácie a nástroje – informácie</b> <b>Výkonový štandard</b> <ul style="list-style-type: none"> <li>posudzovať rôzne reprezentácie pri spracovaní informácií,</li> <li>kódovať informáciu do konkrétnej digitálnej reprezentácie,</li> <li>dekódovať informáciu do konkrétnej digitálnej reprezentácie</li> <li>argumentovať pre voľbu nástrojov.</li> </ul> <b>Obsahový štandard</b> Pojmy: bit, bajt, kibibajt, mebibajt (resp. bit, bajt, kilobajt, megabajt), digitalizácia Vlastnosti a vzťahy: vzťahy medzi jednotlivými typmi informácií (text, čísla) Procesy: kódovanie textovej informácie	<b>Informatické myslenie:</b> <b>Logika</b> <ul style="list-style-type: none"> <li>(LOG4) vyvodzovať (logicky zdôvodňovať) závery z pozorovaní a experimentov (aj myšlienkových)</li> <li>(LOG6) logicky zdôvodniť zmenu algoritmu/programu</li> <li>(LOG 8) z existujúcich pravidiel logicky odvodzovať iné pravidlá</li> </ul> <b>Algoritmy</b> <ul style="list-style-type: none"> <li>(ALG3) vytvárať vlastné algoritmy riešiace problém (návrh vlastného kódovania textu)</li> </ul> <b>Abstrakcia</b> <ul style="list-style-type: none"> <li>(ABS3) využiť podstatné prvky problémov (vytvárať model, vyjadriť pomocou tabuľky)</li> </ul>
<b>Riešený didaktický problém</b>	
Aby si žiaci uvedomili možnosti, ale aj limity či perspektívy súčasných technológií, potrebujú poznať princípy ich fungovania. Voľba vhodnej znakovkej sady je dôležitá aj neskôr, keď žiaci budú vytvárať vlastné webové stránky.	
<i>Dominantné vyučovacie metódy a formy</i>	<i>Príprava učiteľa a pomôcky</i>
<ul style="list-style-type: none"> <li>Bádateľská metóda (model 5E),</li> <li>frontálna a individuálna forma.</li> </ul>	pre učiteľa <ul style="list-style-type: none"> <li>I_SS_56_Kody_sifry_kompresia_M.pdf metodika vyučovania vo formáte pdf</li> <li>I_SS_56_Kody_sifry_kompresia_PL_riesenie.docx</li> <li>I_SS_56_Kody_sifry_kompresia_PL.docx pracovný list a pracovný list s riešenými úlohami</li> <li>mesto.jpg pracovný súbor</li> </ul> pre žiaka <ul style="list-style-type: none"> <li>I_SS_56_Kody_sifry_kompresia_PL.pdf pracovný list</li> </ul> Použitie digitálnych nástrojov: NUTNÉ
<b>Diagnostika splnenia vzdelávacích cieľov</b>	
Sebahodnotiaci test v pracovnom zošite.	

## Úvod

Tretia metodika v sérii Kódy, šifry, kompresia nadväzuje na predchádzajúce. Jej cieľom je ukázať žiakom, akým spôsobom počítač pracuje s textovou informáciou.

Keďže pri kódovaní textovej informácie je potrebné opätovne pracovať s binárnou sústavou, v rámci tejto metodiky si žiaci precvičia zručnosti získané v predchádzajúcej téme – prevody medzi dekadickou a binárnou sústavou. Žiaci sa naučia aj druhý algoritmus prevodu z dekadickej do binárnej sústavy (s využitím celočíselného delenia a zvyšku). Naučia sa používať aplikáciu Kalkulačka v režime Programátorská, ktorá im tieto prevody umožní zrýchliť, prípadne im umožní skontrolovať si svoje výpočty (použitie tejto aplikácie je možné nahradiť niektorou online kalkulačkou).

Žiaci sa v rámci tejto metodiky zoznámia s pojmom bit, keďže potrebujeme budovať aj odbornú terminológiu. Tento pojem využijeme aj pri výpočtoch veľkosti súborov (v tejto metodike textových, neskôr grafických, zvukových,...).

**Poznámka:**

Odporúčame po tejto vyučovacej hodine zaradiť ďalšiu vyučovaciu hodinu, na ktorej žiakom predstavíme napr. znakovú sadu windows-1250, ktorú nájdeme v Učebnici informatiky pre stredné školy v prehľadnej tabuľke. Ak ste za metodikou Ako kóduje počítač – čísla nezaradili vyučovaciu hodinu venovanú iným číselným sústavám (a medzi nimi aj šestnástkovej), môžete do obsahu zaradenej hodiny zahrnúť aj šestnástkovú sústavu (pri práci so znakovou sadou windows-1250 je ju možné efektívne využiť). Za tabuľkou nasledujú úlohy, prostredníctvom ktorých si žiaci môžu precvičiť kódovanie textu.

Táto tabuľka zároveň demonštruje žiakom prehľadnejší zápis priradenia znak – kód, teda nie v lineárnej podobe, ako je to v tejto metodike, ale v dvojrozmernej štruktúre.

Žiaci majú k dispozícii pracovný list, ktorý obsahuje zadania úloh, miesto na žiacke riešenie a miesto pre poznámky. Odporúčame, aby učiteľ žiakom pri každej fáze vyučovania uviedol zoznam úloh z pracovného listu, ktoré budú aktuálne riešiť. Poslednou časťou je sebahodnotiaci test.

**Poznámka:**

Pracovný list je jedným z výstupov žiaka. Odporúčame, aby si žiaci jednotlivé vypracované pracovné listy odkladali. Neskôr ich môžu využiť pri opakovaní učiva.

## PRIEBEH VÝUČBY

Osnova vyučovacej hodiny (podľa modelu 5E):

- **Zapojenie (5 minút)** – diskusia so žiakmi na tému kódovania textu
- **Skúmanie (10 minút)** – skúmanie princípov činnosti mechanických kalkulačiek (úlohy 1 a 2 z pracovného listu)
- **Vysvetlenie (10 minút)** – vysvetlenie predchádzajúcich zistení, riešenie úloh (úlohy 3 až 5 z pracovného listu)
- **Rozpracovanie (6 minút)** – riešenie náročnejších úloh (úlohy 6 až 8 z pracovného listu)
- **Vyhodnotenie (4 minúty)** – vyriešenie sebahodnotiaceho testu, kontrola odpovedí

## ZAPOJENIE (CCA 5 MIN)

Na predchádzajúcej vyučovacej hodine sme si ukázali, akým spôsobom sú v počítači reprezentované číselné údaje – pomocou elektronických súčiastok, ktorými elektrický prúd prechádza alebo neprechádza. Pre našu lepšiu predstavu o tejto reprezentácii si my tieto dva možné stavy kódujeme pomocou číier 0 a 1, a naše „dekadické“ čísla zapisujeme pomocou binárnej číselnej sústavy práve pomocou týchto dvoch číier.

Počítače sa však dávnejšie posunuli od funkcie „rýchlej a presnej kalkulačky“ k všestrannejšiemu použitiu. Prvé pokusy o spracovanie aj iných ako číselných údajov siahajú do 19. storočia a súvisia so sčítaním ľudu v Spojených štátoch amerických v roku 1890. Ako asistent sa ho zúčastnil aj devätnásťročný študent matematiky Herman Holerith. Všimol si, aké obrovské množstvo údajov je potrebných pri takomto podujatí spracovať a aké je to časovo náročné – kým boli výsledky spracované a vyhodnotené, už neboli aktuálne. Pri ďalšom sčítaní ľudu v roku 1890 sa už používal jeho vynález – elektrický tabulačný systém, ktorý dokázal spracovať údaje špeciálne zaznamenané na diernom štítiku desať krát rýchlejšie ako manuálne spracovanie. Asistenti „prepísali“ odpovede každého občana na dierny štítok, ten bol následne použitý ako zdroj údajov pre elektrický tabulačný systém:

1	2	3	4	GM	UM	Jp	Ch	Oc	In	20	50	80	Dv	Un	3	4	3	4	A	E	L	a	g
5	6	7	8	CL	UL	O	M <sub>1</sub>	Qd	Mo	25	55	85	Wd	CY	1	2	1	2	B	F	M	b	h
1	2	3	4	CS	US	Mb	B	M	0	30	60	0	2	Mr	0	15	0	15	C	G	N	c	i
5	6	7	8	No	Hd	W	F		5	35	65	1	3	Sg	5	10	5	10	D	H	O	d	k
1	2	3	4	Fh	Ff	Fm	7	1	10	40	70	90	4	0	1	3	0	2	St	I	P	e	l
5	6	7	8	Hh	Hf	Hm	8	2	15	45	75	95	100	Un	2	4	1	3	4	K	Un	f	m
1	2	3	4	X	Un	Ft	9	3	i	e	X	R	L	E	A	6	0	US	Ir	Sc	US	Ir	Sc
5	6	7	8	Ot	En	Mt	10	4	k	d	Y	S	M	F	B	10	1	Gr	En	Wa	Gr	En	Wa
1	2	3	4	W	R	OK	11	5	l	e	Z	T	N	G	C	15	2	Sv	FC	EC	Sv	FC	EC
5	6	7	8	7	4	1	12	6	m	f	NG	U	O	H	D	Un	3	Nw	Bo	Hu	Nw	Bo	Hu
1	2	3	4	8	5	2	Oc	0	n	g	a	V	P	I	Al	Na	4	Dk	Fr	It	Dk	Fr	It
5	6	7	8	9	6	3	0	p	o	h	b	W	Q	K	Un	Pa	5	Ru	Ot	Un	Ru	Ot	Un

Obrázok 1 - Dierny štítok použitý pri sčítaní ľudu v USA (1890)

<http://www.columbia.edu/cu/computinghistory/1890tabulator.html>

Holleritha inšpirovali sprievodcovia vo vlaku, ktorý špeciálnymi klieštikmi cvikli cestujúcemu papierový lístok, čím zaregistrovali jeho prítomnosť vo vlaku a tiež to, že už bol skontrolovaný.

Skúsme prísť na princíp, ako fungoval **elektrický** tabulačný systém v spolupráci s diernym štítkom. Môžeme si pomôcť tým, že si pozrieme samotný [formulár s otázkami](#).

Žiakom necháme priestor na **hľadanie odpovedí na otázky**:

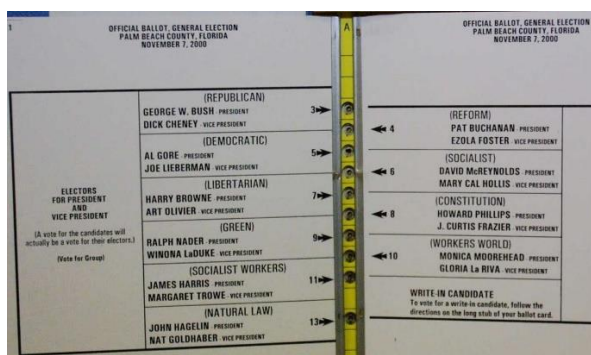
- Akým spôsobom prebiehalo zapísanie odpovede na dierny štítok?
- Ako fungovalo vyhodnotenie dierného štítku?

V diskusii by sme mohli prísť **k týmto záverom**:

- Každéj otázke bola na štítku priradená istá plocha.
- Každéj odpovedi na danú otázku bol priradený konkrétny bod v rámci plochy pre otázku.
- Vyrazením dierky bola zaznamenaná príslušná odpoveď na danú otázku (napr. pohlavie M (male)/F (female), alebo veková kategória).
- Pri hľadaní odpovede na danú otázku (napr. „Koľko je v Spojených štátoch amerických mužov starších ako 60 rokov, ktorý nevedia písať?“) stačilo vhodne nastaviť príslušné elektrické kontakty a triedič štítky podľa tejto otázky dierne štítky roztriedil a sčítal.

Dierny štítok (aj v podobe diernej pásky) slúžil ako pamäťové médium ešte aj v 20. storočí nielen ako nosič údajov, ale aj programového kódu, čo umožnilo uchovávať a opakovane vykonávať vytvorené programy.

Dokonca v roku 2000 bol použitý pri voľbách prezidenta Spojených štátov amerických – aj keď práve jeho použitie bolo príčinou podozrení na nekorektné výsledky volieb. Dierny štítok (žltý pás v strede) mohol zväzdať k nesprávnemu priradeniu dierky ku kandidátovi na prezidenta. Výsledkom týchto volieb bol prezident George W. Bush a koniec volebného lístka v podobe dierného štítku. Nezlyhal dierny štítok, zlyhal človek, ktorý viac ako storočnú technológiu nesprávne aplikoval. Myslíme si, že tento konkrétny, praktický príklad poskytuje žiakom možnosť nahliadnuť do reálneho sveta, kde aj najmenšia chyba môže mať obrovské dôsledky.



Obrázok 2 - Hlasovací lístok v prezidentských voľbách USA (2000)

[https://en.wikipedia.org/wiki/2000\\_United\\_States\\_presidential\\_election\\_recount\\_in\\_Florida#/media/File:Butterfly\\_Ballot,\\_Florida\\_2000\\_\(large\).jpg](https://en.wikipedia.org/wiki/2000_United_States_presidential_election_recount_in_Florida#/media/File:Butterfly_Ballot,_Florida_2000_(large).jpg)

## SKÚMANIE (CCA 15 MIN)

Dierne štítky sú už súčasťou histórie počítačov. Ale poslúžia nám ako odrazový mostík k hľadaniu otázky, ako sú v počítači reprezentované textové (ale aj iné) informácie.

Pri diernom štítku bol jedinečný kód priradený konkrétnej slovnej odpovedi – čiže vidíme tu potrebu previesť textovú informáciu na číselnú (0 – na príslušnom mieste nie je dierka, pri nastavení kontaktu elektrický prúd nebude pretekať; 1 - na príslušnom je dierka, pri nastavení kontaktu elektrický prúd bude pretekať). A takto sme sa opäť dostali k dvojkovej sústave. Celý dierny štítok by sme mohli reprezentovať pomocou postupnosti cifier 0 a 1, ktoré by reprezentovali prítomnosť dierky na príslušnom mieste pre odpoveď, pričom by sme vedeli, ktoré bloky núl a jednotiek prislúchajú jednotlivým otázkam.

Takáto reprezentácia textových odpovedí je jednoúčelová, slúži pre danú situáciu. Skúsme to teda zovšeobecniť tak, aby sa nami navrhnutý systém zápisu textovej informácie dal použiť univerzálne.

**Úloha 1** Kamil navrhol takýto postup reprezentácie veľkých písmen anglickej abecedy:

$A \Rightarrow 0$ ,  $B \Rightarrow 1$ ,  $C \Rightarrow 10$ ,  $D \Rightarrow 11$ ,  $E \Rightarrow 100$ ,  $F \Rightarrow 101$ ,  $G \Rightarrow 110$ ,  $H \Rightarrow 111$ ,  $I \Rightarrow 1000$ ,  $J \Rightarrow 1001$ ,  $K \Rightarrow 1010$ ,  
 $L \Rightarrow 1011$ ,  $M \Rightarrow 1100$ ,  $N \Rightarrow 1101$ ,  $O \Rightarrow 1110$ ,  $P \Rightarrow 1111$ ,  $Q \Rightarrow 10000$ ,  $R \Rightarrow 10001$ ,  $S \Rightarrow 10010$ ,  $T \Rightarrow 10011$ ,  
 $U \Rightarrow 10100$ ,  $V \Rightarrow 10101$ ,  $W \Rightarrow 10110$ ,  $X \Rightarrow 10111$ ,  $Y \Rightarrow 11000$ ,  $Z \Rightarrow 11001$

Je toto kódovanie navrhnuté správne?

Riešenie:

Necháme priestor na vyjadrenia žiakov – zaznamenávame si ich postrehy na tabuľu, aby sme ich mohli potom „zapracovať“ do správneho riešenia. Ak žiaci nedokážu objaviť problematické miesto tohto postupu, uvedieme príklad:

Z predchádzajúcej vyučovacej hodiny vieme, že počítač „vie pracovať“ len s dvoma stavmi: 0 a 1. Preto je jasné, že jednotlivým znakom musíme priradiť postupnosť núl a jednotiek. Dôležité však je, aby sa takto zakódovaná textová informácia dala aj späťne dekódovať. V prípade Kamilovho návrhu to však nie je možné, napr.:

Ak zakódujeme slovo DOM  $\Rightarrow 1111111100$ , získanú postupnosť nedokážeme späťne jednoznačne dekódovať. Nevieme, či osem jednotiek reprezentuje osem písmen B alebo štyri písmená D alebo dve písmená O ... Možných výsledkov je veľa.

Teda potrebujeme do tohto riešenia zakomponovať ešte jeden faktor – a tým je dĺžka kódu (s obdobnou úlohou sa žiaci už stretli v rámci prvej metodiky tejto série). Ak bude dĺžka kódu pre každý znak rovnaká, dokážeme jednoznačne identifikovať príslušný znak.

Upravme teda Kamilov postup. Vidíme, že najdlhšia postupnosť vo vytvorenom systéme je tvorená piatimi nulami a jednotkami. Ako teda upraviť kódy tých znakov, ktoré sú kratšie? Doplníme úvodné nuly – tie nezmenia hodnotu kódu, len upravujú jeho dĺžku:

A	00000
B	00001
C	00010
D	00011
E	00100
F	00101
G	00110

H	00111
I	01000
J	01001
K	01010
L	01011
M	01100
N	01101

O	01110
P	01111
Q	10000
R	10001
S	10010
T	10011
U	10100

V	10101
W	10110
X	10111
Y	11000
Z	11001

Takto zvolené kódovanie daných znakov je efektívne (kratšia postupnosť neexistuje) a je jednoznačné – vieme každému znaku priradiť jedinečný kód, každému kódu jedinečný znak. Vyskúšajme si to:

**Úloha 2** Pomocou upraveného Kamilovho kódovania znakov:

- zakódujte slovo LES,
- zistíte, aké slovo sa skrýva za kódom 110011010000001,
- zistíte, aké slovo sa skrýva za kódom 111001010000001.

Riešenie:

- Výsledný kód: 010110010010010
- Kód si rozdelíme na päťice núl a jednotiek 11001|10100|00001 a každá päťica nám určí znak: ZUB.
- Kód si rozdelíme na päťice núl a jednotiek 11100|10100|00001 a každá päťica nám určí znak. Avšak pri pokuse priradiť kódu 11100 znak z našej tabuľky zistíme, že takéto priradenie neexistuje.

Žiakov sa spýtame, či sú binárne kódy sú jednotlivým znakom priradené nie náhodne alebo systematicky. Navrhujeme im, aby binárne čísla previedli do „našej“, dekadického sústavy. Ukážeme im, ako si prevod urýchliť s využitím aplikácie Kalkulačka v režime Programátorská a rozdelíme im jednotlivé znaky tak, aby nemusel každý žiak prepočítavať všetky čísla (môžeme použiť aj niektorú z online kalkulačiek, napr. <https://prevodyonline.eu/sk/ciselne-sustavy.html>). Získané hodnoty zaznamenáme do tabuľky:

A	00000	0
B	00001	1
C	00010	2
D	00011	3
E	00100	4
F	00101	5
G	00110	6

H	00111	7
I	01000	8
J	01001	9
K	01010	10
L	01011	11
M	01100	12
N	01101	13

O	01110	14
P	01111	15
Q	10000	16
R	10001	17
S	10010	18
T	10011	19
U	10100	20

V	10101	21
W	10110	22
X	10111	23
Y	11000	24
Z	11001	25

Vidíme, že tieto kódy „idú za sebou“, teda nie sú náhodné. Navyše, predsa len sa nám, ľuďom, lepšie manipuluje s číslom zapísaným v desiatkovej sústave, preto je dobré, ak si za binárnou reprezentáciou znaku vieme predstaviť jeho hodnotu v desiatkovej sústave.

## VYSVETLENIE (CCA 10 MIN)

V tejto chvíli je vhodné začať používať vhodné odborné pojmy. Rozšírime si, prípadne upresníme dôležitý pojem, ktorý začneme v nasledujúcich úlohách používať – bit.

**Úloha 3** Využite Slovníkový portál Jazykovedného ústavu Ľudovíta Štúra SAV a zistite, čo znamená slovo „bit“.

Riešenie:

Slovníkový portál Jazykovedného ústavu Ľudovíta Štúra SAV nám prezradí, že bit je:

- dvojková číslica (0, 1)
- v počítačovej pamäti miesto na jednu číslicu v dvojkovej sústave
- jednotka na meranie množstva informácie.

Doplníme túto definíciu o skratku tejto jednotky – b.

Aký to má dopad na našu terminológiu? Demonštrujeme to na už vyriešenej úlohe 1, ktorej závery môžeme preformulovať takto:

- Kamilov navrhovaný systém kódovania sme nahradili 5-bitovým kódovaním.  
alebo
- Navrhnutý kód pre jeden znak zaberie v pamäti počítača 5 bitov.  
alebo
- Navrhnutý kód pre jeden znak zaberie v pamäti počítača 5 b.

**Úloha 4** a) Pomocou našej tabuľky vieme zakódovať „do reči počítača“ len veľké písmená anglickej abecedy. Koľko bitové kódovanie potrebujeme použiť, aby sa nám podarilo doplniť tabuľku z úlohy 1 o kódy pre malé písmená anglickej abecedy?  
b) Koľko znakov dokážeme zakódovať pomocou 8-bitového kódovania?

Riešenie:

Pomôžeme si matematikou, konkrétne kombinatorikou – ak si na ňu žiaci nespomenú, nevadí, vieme si pomôcť schémou:

Pri tvorbe 5-bitového kódu obsadzujeme päť pozícií jednou z dvoch cifier 0 a 1. Pri obsadzovaní každej z pozícií máme teda na výber dve možnosti, čo môžeme rozkresliť pomocou stromového diagramu. Výsledný počet všetkých rôznych postupností núl a jednotiek dĺžky 5 je  $2^5 = 32$ .

Odpoveď teda znie, že pomocou 5-bitového kódu by sme dokázali zakódovať okrem veľkých písmen anglickej abecedy už len sedem písmen malej anglickej abecedy. My potrebujeme zakódovať všetky malé písmená malej abecedy, nielen sedem (spolu 50 znakov).

Očakávame odpoveď, že je potrebné predĺžiť kód na šesť cifier. Vďaka tomu dokážeme zakódovať až  $2^6 = 64$  znakov. Pre potreby podúlohy 4 a) teda musíme zvoliť 6-bitové kódovanie.



V podúlohe 4 b) si musíme uvedomiť, čo sa skrýva pod pojmom „8-bitové kódovanie“ – ide o kód dĺžky osem (pozostávajúci z číslíc 0 a 1). Či už pomocou stromového diagramu alebo nadviazaním na objavy z podúlohy 4 a) odvodíme, že pomocou 8-bitového kódovania dokážeme zakódovať  $2^8 = 256$  znakov.

Koľko znakov dokáže kódovať náš počítač? A teda koľko bitovú znakovú sadu používa? Ak si uvedomíme, že okrem písmen malej a veľkej anglickej abecedy máme interpunkčné znamienka, čísllice (vo význame znakov, nie čísel), nealfanumerické znaky, písmená špecifické pre daný jazyk (š, č, ĺ, ř, ě, ů, ъ, ...), nebude to si malé číslo. Skúsme naše doterajšie postrehy zovšeobecniť.

**Úloha 5** Z predchádzajúcich úloh a pozorovaní vidíme, že narastajúcim počtom znakov rastie aj dĺžka kódu. Ak viem, koľko znakov chcem zakódovať, ako vypočítam dĺžku kódu?

Riešenie:

Hľadať komplexnú odpoveď môžeme pomocou niekoľkých konkrétnych situácií – zvolíme si počet znakov a k nemu zistíme potrebnú dĺžku kódu. Pri voľbe počtu znakov vyberajme čísla, ktoré sú mocninou čísla 2, aj také, ktoré ňou nie sú. Zapište objavené hodnoty na tabuľku do prehľadnej tabuľky tak, aby žiaci hľadaný vzťah uvideli (ak ho pri riešení úlohy 4 neobjavili). Takýmto skúmaním závislosti dôjdeme k záveru, že pre daný počet znakov zistíme najbližšiu väčšiu alebo rovnú mocninu čísla 2. Práve stupeň tejto mocniny hovorí o potrebnej dĺžke kódu. Žiaci v 1. ročníku SŠ ešte nepoznajú exponenciálne či logaritmické funkcie (rovnice), preto sa uspokojíme s týmto popisným vyjadrením.

## ROZPRACOVANIE (CCA 6 MIN)

Takéto vlastné kódovanie textovej informácie si môžeme navrhovať pre našu vlastnú potrebu, a je vhodné na to, aby sme porozumeli princípu tohto kódovania. Počítače však musia používať jednotné postupy pre prevod znaku do binárneho kódu a naopak. Na to slúži tzv. **znaková sada** (character set, charset), ktorá presne definuje priradenie znaku k jeho číselnej reprezentácii.

Medzi prvé znakové sady patrí ASCII (The American Standard Code for Information Interchange). Pomocou nej zakódujeme 128 znakov (čísllice 0 až 9, písmená anglickej abecedy, niektoré základné interpunkčné symboly, niektoré riadiace kódy a znak medzery). Táto znaková sada je 7-bitová.

Neskôr vznikali nové znakové sady, ktoré obsahovali 256 znakov (8-bitové znakové sady), napríklad znaková sada pre západoeurópske jazyky (ISO Latin 1), pre stredoeurópske jazyky (ISO Latin2), vlastnú znakovú sadu zaviedla firma Microsoft špeciálne pre strednú Európu (*windows-1250*).

Ale ani tieto 8-bitové znakové sady nepostačovali. Ich nástupcami sa stali 16-bitové znakové sady, z nich spomenieme znakovú sadu *Unicode*. Táto znaková sada obsahuje znaky pre rôzne národné abecedy (obsahuje  $2^{16} = 65\,536$  znakov). Je to na jednej strane jej výhodou (je univerzálna), na strane druhej aj nevýhodou – mnoho jazykov využíva len malú časť znakovkej sady, nepoužívané znaky zbytočne „zaberajú miesto“. Riešením tejto dilemy je znaková sada UTF-8, ktorá používa variabilnú (premenlivú) dĺžku kódu. Napríklad prvých 128 znakov (ktoré sú spoločné so znakovou sadou ASCII a sú pre všetky krajiny rovnaké) kóduje pomocou 8 bitov, čím je možné znížiť výslednú veľkosť súboru.



**Úloha 6** V Poznámkovom bloku (alebo inom textovom editore – nie procesore) vytvorte textový súbor s textom Ahoj! Súbor uložte. Vypočítajte predpokladanú veľkosť tohto súboru. Porovnajte ju so skutočnou veľkosťou súboru.

Riešenie:

Zistíme, že skutočná veľkosť vytvoreného súboru je 5 B. Čo znamená skratka B? Bit to nie je, ten má skratku b. Ak si ale uvedomíme, že náš textový súbor obsahuje 5 znakov, vyzerá to tak, že 1 znak zaberá v pamäti počítača 1 B. Vieme, že na kódovanie jedného znaku je použitých 8 bitov (keďže ide o znaky, ktoré sa nachádzajú v pôvodnej, ASCII tabuľke). Z toho môžeme odvodiť, že 1 B = 8 bitov. Jednotku B nazývame bajt (resp. byte).

**Úloha 7** V Poznámkovom bloku otvorte textový súbor, ktorý ste vytvorili v úlohe 6. Dopíšte za výkričník znak Ť. Súbor uložte a zistite skutočnú veľkosť súboru. Čo ste zistili? Súbor uložte. Vypočítajte predpokladanú veľkosť tohto súboru. Porovnajte ju so skutočnou veľkosťou súboru.

Riešenie:

Aj keď pribudol jeden znak, veľkosť súboru sa zvýšila na 7 B. Z toho môžeme dedukovať, že na zakódovanie znaku Ť bolo použité 16-bitové kódovanie.

**Úloha 8** Aká bude veľkosť textového súboru (po skúsenostiach z úloh 6 a 7), ktorý vytvoríme v Poznámkovom bloku a bude obsahovať 6 000 znakov A?

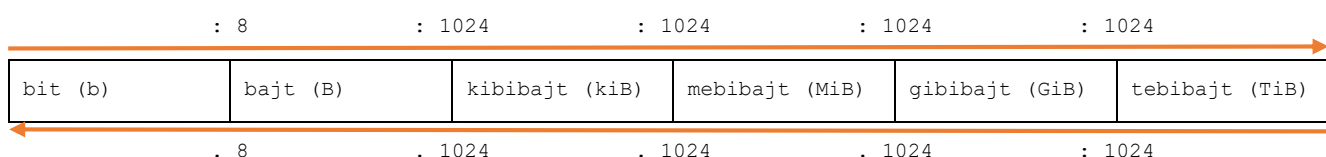
Riešenie

Vieme, že písmeno A je kódované pomocou 8-bitového kódovania. Zaberie teda v pamäti počítača 8 bitov (8 b), čo je 1 bajt (1 B). Ak zapíšeme znak A šesť tisíc krát, súbor zaberie v pamäti počítača  $6\,000 \times 8\text{ b} = 48\,000\text{ b}$ , resp.  $6\,000 \times 1\text{ B} = 6\,000\text{ B}$ . V reálnom živote sa však stretneme s ďalšími jednotkami informácií (a kapacity pamäťových médií).

Samozrejme, ak použijeme textový procesor, veľkosť súboru bude omnoho väčšia napriek tomu, že napíšeme tie isté znaky. Do súboru sa zaznamená aj formát dokumentu, textu, odseku,...

**Poznámka:**

Štátny vzdelávací program uvádza, že máme žiakom uviesť jednotky informácií ako násobky 1000 podľa Medzinárodnej sústavy jednotiek. V tejto metodike sa však prikloníme k štandardizácii IEC 60027-2 a budeme používať binárne predpony a teda násobky 1024 (210). Nechávame na každom učiteľovi, ktorý spôsob si v nasledujúcom texte zvolí.



## VYHODNOTENIE (CCA 4 MIN)

V záverečnej časti hodiny požiadame žiakov, aby vypracovali sebahodnotiaci test. Odporúčame žiakom vysvetliť a zdôrazniť, že cieľom je zistiť čo a ako si žiak z obsahu hodiny zapamätal a nie klasifikácia známku. Pre učiteľa a žiaka zvlášť, je cenná pravdivá informácia o úrovni osvojených poznatkov než umelo vylepšená. Odporúčame žiakom poskytnúť spätnú väzbu ohľadom správnosti odpovedí. Problematické odpovede môžeme so žiakmi prediskutovať na konci vyučovacej hodiny.

### Sebahodnotiaci test

1.	Ako sa zmení počet znakov, ktoré môžeme zakódovať danou znakovou sadou, ak zväčšíme dĺžku binárneho kódu pre jeden znak o jeden bit? a) zdvojnásobí sa                      b) zväčší sa o 2                      c) zväčší sa o 4
2.	Navrhnete znakovú sadu pre kódovanie znakov +, -, *, /, ^, ., (, ), = (symbol čiarky je len oddeľovačom daných znakov). <i>Riešenie:</i> <i>Napr.: Počet znakov je 9. Využijeme vzťah <math>2^3 &lt; 9 &lt; 2^4</math>, z čoho zistíme, že potrebujeme použiť 4-bitové kódovanie, napr.:</i> + => 0000, - => 0001, * => 0010, / => 0011 ^ => 0100, . => 0101, ( => 0110, ) => 0111, = => 1000
3.	Súbor má veľkosť 7 830 B. Jeho veľkosť vyjadrená v kibiBajtoch je a) 7.83 kiB                      b) 7,64 kiB                      c) 62 640 kiB

Odporúčame, aby učiteľ uviedol správne odpovede a na záver zhrnul nové poznatky v zmysle:

- Podobne ako číselnú informáciu, aj textovú kódujeme pomocou binárneho kódu.
- Dĺžka použitého kódu závisí od počtu znakov, ktoré potrebujeme zakódovať.
- Dĺžku kódu môžeme opísať aj použitím jednotky bit – ak je dĺžka kódu  $n$ , hovoríme o  $n$ -bitovom kódovaní.
- Bit predstavuje miesto v pamäti počítača, na ktoré sa zaznamená informácia v podobe dvojkovej cifry 0 alebo 1. Bit teda používame aj ako jednotku informácií.
- To, ako sa jednotlivé znaky kódujú, určuje znaková sada (napr. ASCII, windows-1250, Unicode či UTF-8).
- Používame aj ďalšie jednotky informácií – bajt (B), kibibajt (kiB), mebibajt (MiB), gibibajt (GiB), tebibajt (TiB). Pri prevode medzi týmito jednotkami sa môžeme oprieť o pomocnú tabuľku:

: 8		: 1024		: 1024		: 1024		: 1024	
bit (b)	bajt (B)	kibibajt (kiB)	mebibajt (MiB)	gibibajt (GiB)	tebibajt (TiB)				
: 8		: 1024		: 1024		: 1024		: 1024	

- Odporúčame žiakom ponúknuť video <https://youtu.be/6auO8djKRrM>, ktoré je zaujímavým zakončením dnešnej témy.
- Kde sa na klávesnici „skrývajú“ ostatné znaky znakovkej sady? Vyskúšajte v textovom editore podržať stlačený kláves ALT a k nemu rôzne kombinácie na numerickej klávesnici (prehľad

niektorých nájdete napr. na <https://www.zn.sk/tabulka-znakov-a-specialnych-znakov-klavesove-skratky-cez-alt/>).