



1.1 Informatika obecně

Co si představíš pod pojmem informatika?

Co to informatika podle tebe je?

Kde se s informatikou dneska můžeš setkat?

Kde se znalosti z informatiky využívají?

Co zkoumají vědci zabývající se informatikou?



Léto teenagerů 2024
(8. až 12. července)

Informatika trochu jinak

Jaká informatická témata by tě zajímala?
O čem by ses chtěl dozvědět více?

1.

2.

3.

4.

Jak nás informatika ovlivňuje?



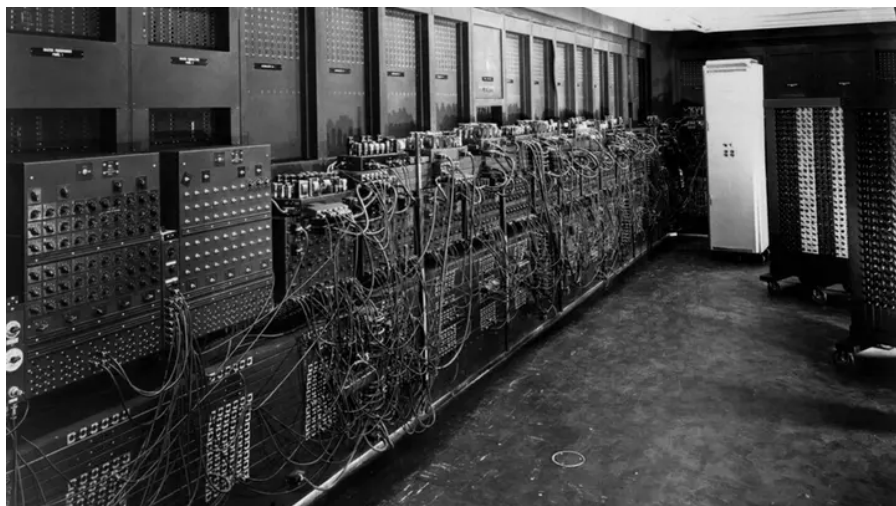
1.2 Instrukce

Instrukce na ovládání člověka:	
Název instrukce	Popis instrukce
1.	
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
10.	
11.	
12.	
13.	
14.	
15.	

1.3 Úvod k počítačům

Počítač je **stroj**, který plní nějaké námi vymyšlené **úkoly** (něco spočítej, zobraz něco na obrazovce, ...), které zapíšeme jako **příkazy**.

První **elektronické počítače** tak jak je známe dnes vznikali v druhé polovině 20. století nicméně jakési představy počítače se objevovali i v předchozích stoletích, tehdy ale lidé ještě neznali **elektronické součástky** jako jsou diody, tranzistory a elektronky. Nebojte, nemusíme přesně rozumět co tyhle slova znamenají, jednoduše technologie nebyly ještě tak daleko aby lidé dokázali sestavit počítače tak, jak se známe dnes. Snažili se o sestavení počítačů mechanických (třeba i s parním pohonem).



Obrázek 1: Sálový počítač ENIAC

Elektronické počítače začínali jako obrovské stroje zabírající plochu celých pater domů, tzv. **sálové počítače** vznikali od poloviny 20. století. Postupným **zlepšováním elektronických součástek**, zejména jejich **zmenšováním**, nás dovedlo až do stavu, kdy je celý počítač velký jako **placička** co se nám vleze do kapsy (a zároveň je tahle placička nesrovnatelně výkonnější než sálové počítače minulosti).

Základními součástmi dnešních počítačů jsou **procesor**, **základní deska**, **grafická karta**, **operační paměť**, **trvalá paměť** a **zdroj**. Důležité jsou i zařízení, která do počítače připojujeme (např. pomocí USB-A nebo HDMI konektoru) a která rozšiřují funkčnost počítače (např. obrazovka, klávesnice), těmto zařízením říkáme **vstupní a výstupní zařízení**.



K čemu nám ale vlastně počítače jsou?
Jakou plní funkci?

Z jakých (základních) částí se počítače skládají?

Název komponenty	K čemu slouží?
1.	
2.	
3.	
4.	
5.	
6.	



1.4 Programovací jazyky

Každý počítač **ovládáme příkazy**. Existují **různé úrovně složitosti** a **různé sady** těchto příkazů. Platí, že složitější příkazy jsou **přeloženy** na ty jednodušší a těm nejjednodušším říkáme **instrukce**. Jsou to **elementární úkony procesoru** jako je sečtení dvou čísel nebo uložení výsledku do paměti počítače.

Programovací jazyky pracují s příkazy, které jsou **složitější**, jsou tedy překládány na ty jednodušší (na instrukce). Programovací jazyky slouží k psaní **kódu programu**, který má plnit nějaký úkol. **Různé programovací jazyky** se hodí k **různým úkolům**. Některé se používají k programování aplikací pro Windows (C#), jiné zase pro aplikace na telefonu (Swift, Kotlin) či pro webové aplikace (JavaScript, TypeScript, Python). **Liší se svými funkcemi** a proto **nepoužíváme jeden univerzální** jazyk na vše. Obecně si jsou ale programovací jazyky celkem **podobné** a proto, když se naučíme jeden bude pro nás **jednodušší** se naučit další pro jiné úkoly.

```
n = int(input("Zadejte číslo: "))
total_sum = 0
for i in range(1, n + 1):
    total_sum += i
print(f"Součet čísel od 1 do {n} je {total_sum}")
```

Obrázek č.2: Ukázka Python kódu

My si během týdne ukážeme programování v **Pythonu**, co je dnes velmi používaný programovací jazyk, zejména při vývoji webových aplikací nebo při zpracování dat.



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 uloženy 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 tyto 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	2^0
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 **stejně číslo**, zkuste to pro ověření).



Příklady:

$$\begin{array}{rclcl} 1) & 0100 & 0101 & 2) & 1001 & 0000 & 3) & 1100 & 1010 & 4) & 0000 & 1101 \\ + & 0011 & 0100 & + & 0010 & 1111 & + & 1001 & 0010 & + & 1100 & 1101 \\ \hline & 0111 & 1001 & & 1011 & 1111 & & 1 & 0101 & & 1101 & 1010 \end{array}$$

Příklady k procvičení sčítání:

$$\begin{array}{rclcl} 1) & 0000 & 1100 & 2) & 0100 & 0001 & 3) & 0011 & 0010 & 4) & 0001 & 0001 \\ + & 0000 & 0000 & + & 0010 & 1001 & + & 0001 & 0010 & + & 1101 & 0111 \\ \hline & & & & & & & & & & & \end{array}$$

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	Invertované číslo v binární soustavě	Číslo s přičtenou jedničkou
-2			
-10			



Léto teenagerů 2024
(8. až 12. července)

Informatika trochu jinak

Ted' si to vše **spojíme dohromady**, čísla v závorkách nejdříve **převeďte** do binární soustavy a následně **spočtete** 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 (@, \$, %, ...). Tedy každé číslo v této tabulce znamená nějaké písmeno či znak. Slova a věty jsou tedy representována čísla (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).

ASCII Tabulka

0	NUL	16	DLE	32	SPC	48	0	64	@	80	P	96	`	112	p
1	SOH	17	DC1	33	!	49	1	65	A	81	Q	97	a	113	q
2	STX	18	DC2	34	"	50	2	66	B	82	R	98	b	114	r
3	ETX	19	DC3	35	#	51	3	67	C	83	S	99	c	115	s
4	EOT	20	DC4	36	\$	52	4	68	D	84	T	100	d	116	t
5	ENQ	21	NAK	37	%	53	5	69	E	85	U	101	e	117	u
6	ACK	22	SYN	38	&	54	6	70	F	86	V	102	f	118	v
7	BEL	23	ETB	39	'	55	7	71	G	87	W	103	g	119	w
8	BS	24	CAN	40	(56	8	72	H	88	X	104	h	120	x
9	HT	25	EM	41)	57	9	73	I	89	Y	105	i	121	y
10	LF	26	SUB	42	*	58	:	74	J	90	Z	106	j	122	z
11	VT	27	ESC	43	+	59	;	75	K	91	[107	k	123	{
12	FF	28	FS	44	,	60	<	76	L	92	\	108	l	124	
13	CR	29	GS	45	-	61	=	77	M	93]	109	m	125	}
14	SO	30	RS	46	.	62	>	78	N	94	^	110	n	126	~
15	SI	31	US	47	/	63	?	79	O	95	_	111	o	127	DEL

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



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 *****
-------------------	-------------------	-------------------	-------------------	-------------------