

ZPRO 3. cvičení

# Opakování z minula

- Funkce
- Více zdrojových souborů

Největší společný dělitel

# Euklidův algoritmus

## Algoritmus [\[ editovat \]](#) [\[ editovat zdroj \]](#)

---

Mějme dána dvě přirozená čísla, uložená v proměnných  $u$  a  $w$ .

Dokud  $w$  není nulové, opakuj:

Do  $r$  ulož zbytek po dělení čísla  $u$  číslem  $w$

Do  $u$  ulož  $w$

Do  $w$  ulož  $r$

Konec algoritmu, v  $u$  je uložen největší společný dělitel původních čísel.

# Cvičení

- Napište funkci, která vrátí maximum ze 4 zadaných hodnot (parametrů).

**int maximum(int a, int b, int c, int d)**

# Cvičení

- Napište funkci, která vrátí maximum ze 4 zadaných hodnot (parametrů).

**int maximum(int a, int b, int c, int d)**

```
int maximum(int a, int b, int c, int d)
{
    return a >= b ? c >= d ? a >= c ? a : c : a >= d ? a : d : c >= d ? b >= c ? b : c : b >= d ? b : d;
}
```

# Alternativní jméno typu

- `typedef Typ Název_Typu`
- `using Název_Typu = Typ`
  
- `typedef int cislo;`
- `using cislo = int;`
  
- Oba příkazy vytvoří alternativní název (**cislo**) typu **int**

# Celočíselné datové typy

Jméno	Velikost	Nejmenší hodnota	Největší hodnota
short	2 B	−32 768	32 767
int	4 B	−2 147 483 648	2 147 483 647
long	4 B	−2 147 483 648	2 147 483 647
long long	8 B	−9 223 372 036 854 775 808	9 223 372 036 854 775 807

Jméno	Velikost	Nejmenší hodnota	Největší hodnota
unsigned short	2 B	0	65 535
unsigned	4 B	0	4 294 967 295
unsigned long	4 B	0	4 294 967 295
unsigned long long	8 B	0	18 446 744 073 709 551 615



# Reálná čísla

Jméno	Velikost	Rozsah		Počet platných cifer
float	4 B	$3,4 \times 10^{-38}$	$— 3,4 \times 10^{+38}$	7 – 8
double	8 B	$1,7 \times 10^{-308}$	$— 1,8 \times 10^{+308}$	15 – 16
long double	10 B	$3,4 \times 10^{-4932}$	$— 1,1 \times 10^{+4932}$	18 – 19

# Přetypování datových typů

# Zaokrouhlovací chyby

- Jen konečný počet cifer
- Jen konečný počet zobrazitelných čísel
- Ostatní hodnoty se zaokrouhlí na nejbližší zobrazitelné číslo
- ve dvojkové soustavě je  $0,1$  periodické – nelze zobrazit přesně

# Knihovny limits.h a limits

- limits.h resp. climits:
  - INT\_MAX
  - LLONG\_MIN,
  - SHRT\_MAX
- limits
  - `cout << numeric_limits<int>::min() << endl;`
  - `cout << numeric_limits<double>::max() << endl;`
  - `cout << numeric_limits<float>::epsilon() << endl;`

# Aritmetické operace

+ Unární plus (identita)

+ Sčítání

\* Násobení

% Zbytek po dělení (modulo)

– Unární minus (otočení znaménka)

– Odečítání

/ Dělení

++ Inkrementace (zvýšení hodnoty o 1)

-- Dekrementace (snížení hodnoty o 1)

# Knihovna cmath

<code>abs (x)</code>	jež vrací	$ x $ ,	<code>atan (x)</code>	jež vrací	$\operatorname{arctg} x$ ,
<code>sin (x)</code>	jež vrací	$\sin x$ ,	<code>cos (x)</code>	jež vrací	$\cos x$ ,
<code>pow (x, y)</code>	jež vrací	$x^y$ ,	<code>sqrt (x)</code>	jež vrací	$\sqrt{x}$ ,
<code>exp (x)</code>	jež vrací	$e^x$ ,	<code>log (x)</code>	jež vrací	$\ln x$ .

Dále je k dispozici funkce `atan2 (x, y)`, jež vrací  $\operatorname{arctg} x/y$ , a `log10 (x)`, jež vrací desítkový logaritmus  $x$  (v matematice zapisovaný  $\log x$ ).

# Operace s logickými proměnnými

Pro typ `bool` definovány operátory, které vyjadřují následující logické operace:

`&&` pro konjunkci (logický součin),

`||` pro disjunkci (logický součet),

`!` pro negaci.

<b>A</b>	<b>B</b>	<b>A &amp;&amp; B</b>	<b>A    B</b>	<b>!A</b>
false	false	false	false	true
false	true	false	true	true
true	false	false	true	false
true	true	true	true	false

# Znaky

- Klíčové slovo **char** – velikost **1 B**
- **char znak = 'a';**
- Ve skutečnosti se jedná o celočíselný datový typ
- Hodnoty -128 až 127



# Příklad

- Napište funkci **char na\_velke(char a);**

Zápis	Význam	Zápis	Význam
\n	Nová řádka	\t	Horizontální tabulátor
\v	Vertikální tabulátor	\r	Počátek řádky
\f	Konec stránky	\b	Krok zpět
\a	Zvukové znamení	\\	Obrácené lomítko
\'	Apostrof	\"	Uvozovka
\?	Otazník	\0	Znak s kódem 0
\ooo	Znak s kódem <i>ooo</i> (v osmičkové soustavě)	\xhhhh	Znak s kódem <i>hhhh</i> (v šestnáctkové soustavě)

# ASCII tabulka

0	NUL	16	DLE	32		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

# Třída String

- Metody:
  - size(), length() – délka stringu
  - at(i) – vrátí char na i-té pozici

Alternativa k at(i) je přetížený operátor []