

# **Programování v jazyce C pro chemiky** (C2160)

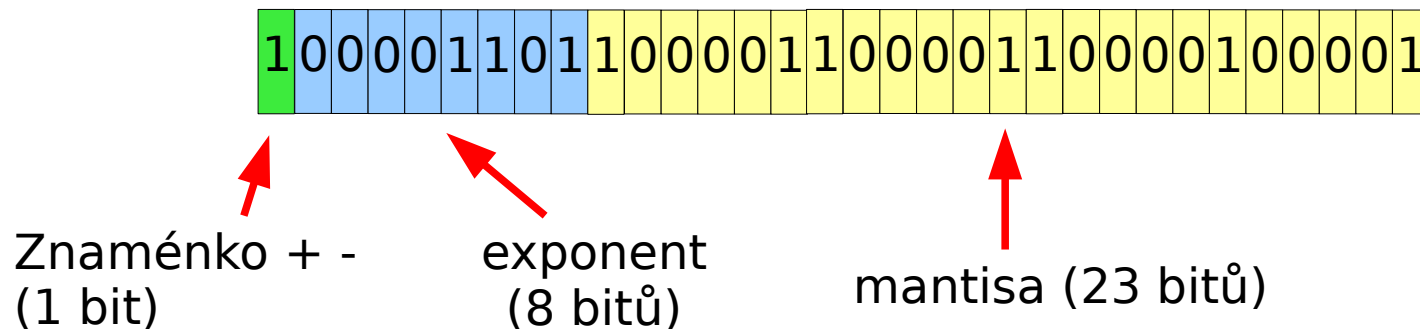
## **12. Specifické problémy při vývoji vědeckého softwaru**

# Reprezentace reálných čísel v počítači

- Reálná čísla jsou v počítači reprezentována jako čísla tvaru  $\pm x \cdot b^e$ , kde  $b$  je číselná soustava (u počítačů dvojková), číslo  $x$  se nazývá **mantisa** a nabývá hodnot mezi 1 až  $b$  a  $e$  je **exponent** (v desítkové soustavě by číslo vypadalo např.:  $1.45 \cdot 10^5$ ,  $8.278 \cdot 10^{-12}$ )
- Hodnoty reálných čísel jsou v počítači reprezentovány určitým počtem bitů, které jsou rozděleny do třech kategorií: znaménko, exponent a mantisa
- Hodnoty mohou nabývat pouze určité velikosti, která je dána počtem bitů použitých pro uložení hodnoty v paměti:

float	32 bit	$1,2 \cdot 10^{-38}$ až $3,4 \cdot 10^{38}$	7 deset. míst
double	64 bit	$2,2 \cdot 10^{-308}$ až $1,8 \cdot 10^{308}$	15 deset. míst

Reprezentace hodnoty typu float v počítači (celkem 32 bitů):

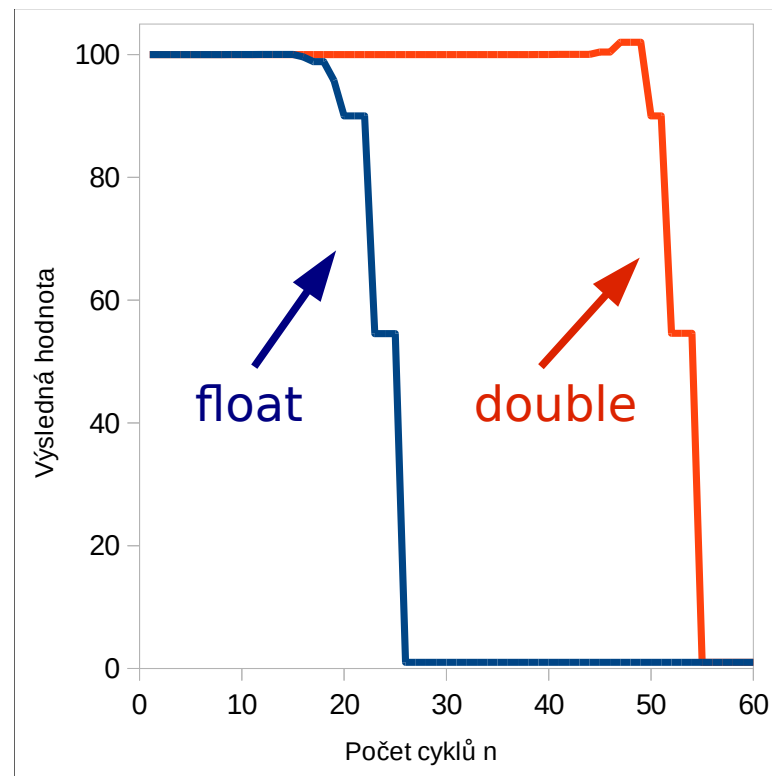


# Přesnost výpočtů

- Reálná čísla jsou v počítači uložena s omezenou přesností, která je dána počtem bitů použitým pro kódování čísla
- Čísla typu double mají větší přesnost než float, proto jsou přednostně používána pro numericky náročné výpočty

```
//Program provede n-krat  
//odmocneni a pak n-krat  
//umocneni  
  
int i = 0;  
int n = 60;  
float a = 100.0;  
  
for (i = 0; i < n; i++)  
    a = sqrt(a);  
  
for (i = 0; i < n; i++)  
    a = a*a;  
  
printf("%f\n", a);
```

```
0 10.0000000  
1 3.1622777  
.....  
5 1.0746078  
.....  
10 1.0022511  
.....  
15 1.0000702  
.....  
20 1.0000021  
21 1.0000010  
22 1.0000004  
23 1.0000002  
24 1.0000001  
25 1.0000000
```



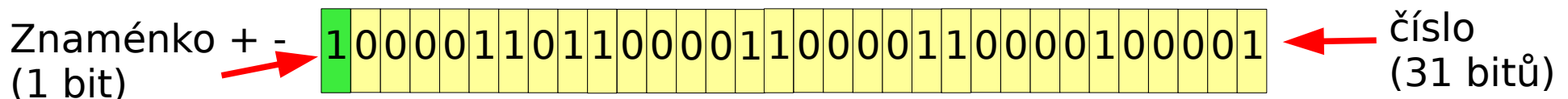
# Hodnoty inf, -inf, nan

- Při aritmetických operacích s reálnými čísly může být výsledkem hodnota, která je vyšší než maximální hodnota pro danou proměnnou (což je např.  $1,8 \cdot 10^{308}$  pro čísla typu double)
- V takovém případě je do proměnné uložena speciální sekvence bitů, která reprezentuje kladné nebo záporné nekonečno
- Při výpisu této hodnoty pomocí funkce `printf()` se na obrazovky místo čísla vypíše text: `inf` nebo `-inf` (některé programovací jazyky mohou používat výpisy: 'infinity', 'INF', 'INFINITY', atd.)
- Není-li výsledek aritmetické operace definován (např. `sqrt(-1)`, `log(-1)`) uloží se do výsledné proměnné sekvence reprezentující nečíselnou hodnotu (angl. *not a number*, zkracuje se *NaN*)
- Při výpisu této hodnoty pomocí funkce `printf()` se na obrazovky místo čísla vypíše text: `nan` (některé programovací jazyky mohou používat výpisy: 'NaN', 'NaN%', 'NaNQ', 'NaNS', 'qNaN', 'sNaN', '1.#SNAN', atd.)
- NaN se chová “divně”: nerovná se ničemu, ani samo sobě (`x == x` je False pro NaN), šíří se (jakákoli operace nad NaN produkuje NaN)

# Reprezentace celých čísel, *integer overflow*

- Celočíselné proměnné mají omezený rozsah, pro 32-bitové číslo typu `int` je rozsah hodnot:  $-2\,147\,483\,648$  až  $2\,147\,483\,647$
- Při překročení maximální hodnoty dochází ke změně bitu reprezentující znaménko a kladné číslo se tak změní na záporné (a opačně) – tento stav se nazývá přetečení (*integer overflow*)

Reprezentace hodnoty typu `int` v počítači (celkem 32 bitů):



$a = 1$	
$a + 1 = 2$	
$a + 2 = 3$	
$a = 2\,147\,483\,646$	
$a + 1 = 2\,147\,483\,647$	
$a + 2 = -2\,147\,483\,648$	
$a + 3 = -2\,147\,483\,647$	

# Celá čísla

- Ke vzniku *přetečení* může snadno dojít např. při násobení, kdy vynásobení relativně malých čísel poskytuje velká čísla
- Příklad (operace prováděny s 32-bit hodnotami typu `int`):

Násobení	výsledek	správná hodnota	
10000 * 10000 =	100000000	100000000	
20000 * 20000 =	400000000	400000000	
30000 * 30000 =	900000000	900000000	
40000 * 40000 =	1600000000	1600000000	
50000 * 50000 =	-1794967296	2500000000	} Integer overflow
60000 * 60000 =	-694967296	3600000000	
70000 * 70000 =	605032704	4900000000	
80000 * 80000 =	2105032704	6400000000	
90000 * 90000 =	-489934592	8100000000	

# Pomocná makra a funkce

- V jazyce C jsou definovány symbolické konstanty (makra) pro zjištění maximálních hodnot pro dané typy:
- `INT_MIN`, `INT_MAX` obsahují nejnižší a nejvyšší možnou hodnotu čísel typu `int` (je třeba vložit hlavičkový soubor *limits.h*)
- `FLT_MAX`, `DBL_MAX` obsahují nejvyšší možnou hodnotu čísel typu `float` a `double` (je třeba vložit hlavičkový soubor *float.h*)
- Funkce `isnan()` a `isinf()` zjišťují, zda-li je daná hodnota NaN nebo nekonečno (`inf`, `-inf`) (je třeba vložit hlavičkový soubor *math.h*)
- Více informací na:  
<https://en.wikipedia.org/wiki/Limits.h>  
<https://en.wikipedia.org/wiki/Float.h>  
Manuálové stránky: *man isinf*, *man isnan*

# Nejdražší počítačová chyba

- Havárie 1. startu rakety Ariane 5 (4. června 1996)
- Chyba v navigačním softwaru: při **konverzi z reálné 64-bitové hodnoty do 16-bit celočíselné** byla hodnota větší než je maximální hodnota pro danou celočíselnou proměnnou, což vedlo k selhání řídicího počítače
- Záložní počítač selhal ve stejném okamžiku v důsledku stejné softwarové chyby
- Použitý software byl původně vyvinut pro starší raketu Ariane 4, která měla jiný profil letu a hodnoty zpracovávané softwarem byly nižší
- Ke zničení rakety došlo 37 sekund po startu
- Více na:  
[http://en.wikipedia.org/wiki/Ariane\\_5\\_Flight\\_501](http://en.wikipedia.org/wiki/Ariane_5_Flight_501)  
[http://en.wikipedia.org/wiki/List\\_of\\_software\\_bugs](http://en.wikipedia.org/wiki/List_of_software_bugs)





# Objektové programování

```
class Atom
{
public:
    Atom();
    void readLine(const string &line, int lineNumber);
    void writeLine(ofstream &ofile);
    void print() const;
    string getResidueName() const;
    int getResidueNumber() const;
private:
    int atomNumber;
    string atomName;
    char alternateLocation;
    string residueName;
};

void Atom::writeLine(ofstream &ofile)
{
    ofile << "ATOM  ";
    ofile << right << setw(5) << atomNumber;
    ofile << left << setw(4) << atomName;
    ofile << right << setw(4) << residueNumber;
    ofile << right << fixed << setprecision(3);
    ofile << setw(8) << coordX;
}
```

# Kurz Pokročilé programování v jazyce C pro chemiky C3220

- Základy objektového programování v C++
- Formátovaný vstup a výstup na obrazovku a do souboru
- Načtení a zápis PDB souboru
- Základy návrhu interaktivních grafických programů s knihovnou Qt

