

POČÍTAČOVÉ CVIČENÍ 3

Cílem cvičení je procvičit si práci s výrazy a operátory a seznámit se s konverzí v jazyce C.

1. Vytvořte program, který vypočte hodnotu exponenciální funkce $y = e^x$ bez použití knihovnických matematických funkcí s využitím rozvoje v Taylorovu řadu (viz Matematika I):

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!} - \dots = \sum_{n=0}^{\infty} \frac{x^n}{n!} \quad (1)$$

Vstupní parametr x vygenerujte jako číslo sestávající se z náhodně vygenerované celé části v rozsahu $<0; 9>$ a desetinné části (dvě místa) získané náhodným vygenerováním celého čísla v rozsahu $<0; 99>$. V konzolovém okně vytiskněte pro kontrolu a srovnání i hodnotu exponenciální funkce vstupního parametru vypočítanou funkcí `exp()` z knihovny `math.h`. Dále sestavte algoritmus, který bude postupně přidávat do výpočtu exponentu členy Taylorovy řady a výsledek vždy s uvedením počtu započítaných členů zobrazí (viz obr. 1). Připočítání a tisk výsledku provádějte v cyklu `for`. Pro dosažení velmi přesného výsledku je potřeba pro daný rozsah vstupního parametru asi padesát členů Taylorovy řady. Výsledky exponenciální funkce zobrazujte s 22 platnými a 16 desetinnými čísly. Proměnné výpočtu exponenciální funkce deklaruje jako `float`, `double`, `long double` a pozorujte dosažitelnou přesnost výpočtu (u tisku `long double` funkcí `printf()` vložte těsně před specifikátor `f` velké písmeno `L`, tedy `%22.16Lf`. Úvodní část zdrojového kódu programu k příkladu je v `C03_1.cpp`.

2. Vytvořte program, ve kterém naplníte dvě šestnáctiprvková pole `a_bin` a `b_bin` náhodně hodnotami 0 a 1. Tato čísla a jejich jednotlivé prvky budou představovat jednotlivé bity binárních čísel (16 bitových bez znaménka). Napište algoritmus, který tato binární čísla převede na dekadická (`a_dec`, `b_dec`), a proveďte postupně všechny operace, které lze dělat na bitové úrovni s dekadickými čísly. Odpovídající operace rovněž proveďte s binárními poli a výsledky vzájemně konfrontujte. Pro úplnost dodávám požadované operace: konjunkce po bitech, disjunkce po bitech, nerovnost po bitech, bitový komplement, bitový posuv (stačí vpravo o náhodný počet bitů v rozsahu 1 až 8). Řešení je zřejmé z obrázku 2, do proměnné `c_bin` se ukládají výsledky operací nad binárními poli, které se převedou do dekadické proměnné `c_dec`, a do proměnné `d_dec` se pak ukládají výsledky požadovaných operací s dekadickými čísly `a_dec` a `b_dec`. Úvodní část zdrojového kódu programu k příkladu je v `C03_2.cpp`.
3. Tímto příkladem si ukážeme, že „C-čko“ nemusí být nuda ani v konzolové aplikaci, budete totiž tvořit vaši zřejmě první hru. Ve zdrojovém kódu programu k příkladu `C03_3.cpp` je prakticky vše připraveno, na vás je dopsat jediný řádek označený přemi tečkami. Cílem je naprogramovat klasickou hru, kdy se snažíte vrhem nebo střelou zasáhnout cíl. Jádrem je klasická úloha pohybové rovnice pro šikmý vrh a tu musíte naprogramovat. V programu jsou pro vás stěžejní proměnné `ele` (definuje elevační úhel hodu ve stupních) a `vel` (definuje počáteční rychlost v m/s). Vaším úkolem je ve smyčce `for` pro vstupní polohovou proměnnou `x` (vodorovná osa, resp. vzdálenost od počátku odkud je hod prováděn) sestavit rovnici, která vypočte hodnotu polohové souřadnice `y` (výšku střely či kamene) pro daný elevační úhel a počáteční rychlost. Polohové souřadnice jsou definovány v metrech.

```

c:\d:\Dokumenty\Visual Studio 2008\Projects\C03_1\Debug\C03_1.exe
Exponential function of 8.41 using math library is 4491.7605115486886000

Taylor series application:

1 term: 1.0000000000000000
2 terms: 9.4100000000000001
3 terms: 44.7740500000000030
4 terms: 143.9112701666666700
5 terms: 352.3472755670033300
6 terms: 702.9366366505842100
7 terms: 1194.3460577692913000
8 terms: 1784.7393765704805000
9 terms: 2405.3903529602308000
10 terms: 2985.3542097866530000
11 terms: 3473.1038133776742000
12 terms: 3846.0105557595366000
13 terms: 4107.3560310454923000
14 terms: 4276.4264500574063000
15 terms: 4377.9894660495629000
16 terms: 4434.9324636824986000
17 terms: 4464.8631268133104000
18 terms: 4479.6700019268474000
19 terms: 4486.5881030215614000
20 terms: 4489.6502730324319000
21 terms: 4490.9379155220031000
22 terms: 4491.4535856809216000
23 terms: 4491.6507123189449000
24 terms: 4491.7227921026742000
25 terms: 4491.7480500602230000
26 terms: 4491.7565468371413000
27 terms: 4491.7592952176756000
28 terms: 4491.7601512873161000
29 terms: 4491.7604084139475000
30 terms: 4491.7604829806705000
31 terms: 4491.7605038842084000
32 terms: 4491.7605095551362000
33 terms: 4491.7605110455270000
34 terms: 4491.7605114253511000
35 terms: 4491.7605115193019000
36 terms: 4491.7605115410764000
37 terms: 4491.7605115471506000
38 terms: 4491.7605115483493000
39 terms: 4491.7605115486149000
40 terms: 4491.7605115486722000
41 terms: 4491.7605115486840000
42 terms: 4491.7605115486867000
43 terms: 4491.7605115486876000
44 terms: 4491.7605115486876000
45 terms: 4491.7605115486876000
46 terms: 4491.7605115486876000
47 terms: 4491.7605115486876000
48 terms: 4491.7605115486876000
49 terms: 4491.7605115486876000
50 terms: 4491.7605115486876000

```

Obr. 1. Příklad výstupů v konzolovém okně pro výpočet exponenciální funkce.

```

c:\d:\Dokumenty\Visual Studio 2008\Projects\C03_2\Debug\C03_2.exe
Two binary inputs and their decimal values:
a_bin: 0 0 1 0 0 0 0 0 0 1 0 0 0 1 1      a_dec: 8227
b_bin: 1 1 1 1 1 0 1 1 1 0 0 0 1 0 1 1      b_dec: 64395

Bit-wise conjunction
c_bin: 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1      c_dec: 8195      d_dec: 8195

Bit-wise disjunction
c_bin: 1 1 1 1 1 0 1 1 1 0 1 0 1 0 1 1      c_dec: 64427     d_dec: 64427

Bit-wise inequality
c_bin: 1 1 0 1 1 0 1 1 1 0 1 0 1 0 0 0      c_dec: 56232     d_dec: 56232

Bit shift to right: 4 bits of a_bin/a_dec
c_bin: 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0      c_dec: 514       d_dec: 514

Bit-wise complement to a_bin/a_dec
c_bin: 1 1 0 1 1 1 1 1 1 0 1 1 1 0 0      c_dec: 57308     d_dec: 57308_

```

Obr. 2. Výstupy binárních operací z příkladu 2 v konzolovém okně.

$$y = x \cdot \operatorname{tg} \alpha - \frac{1}{2} \cdot g \cdot x^2 \cdot \left(\frac{1}{v_0 \cdot \cos \alpha} \right)^2 \quad (2)$$

4. Neveřejný příklad zadaný cvičícím.

```
d:\Dokumenty\Visual Studio 2008\Projects\C03_3\Debug\C03_3.exe
```

Angle: 44 deg., velocity: 29m/s

!!!BLACK HOWK DONE - URRRAAA!!! YOU NEEDED 11 SHOTS

The image shows a classic physics simulation output. A series of white circles forms a smooth, symmetric arc starting from the bottom left, reaching its peak near the center, and ending at the bottom right. The circles are more densely packed at the beginning and end of the path.

Press SPACE + ENTER

Obr. 3. Konzolová střílečka z příkladu 3.