

# Poslední nenulová číslice faktoriálu

Kateřina Bambušková  
BAM015, I206

## Abstrakt

V tomto článku je popsán a vyřešen problém s určením poslední nenulové číslice faktoriálu přirozeného čísla  $N$ . Celý princip spočívá ve vyřešení základních problémů, v návržení vývojového diagramu a jednoduchého algoritmu založeného na iterační metodě výpočtu faktoriálu. Na závěr jsou uvedeny dvě implementace navrženého algoritmu v programovacím jazyce Visual Basic a C++.

## 1. Úvod

Faktoriál čísla  $n$ , značíme jako  $n!$  s vykřičníkem tj.  $n!$  Faktoriál čísla je funkce definovaná jako součin všech čísel od 1 po číslo  $n$  nebo lze i součin všech čísel od  $n$  do 1, i zde totiž platí komutativní zákon.  $n! = n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot 2 \cdot 1 = 1 \cdot 2 \cdot \dots \cdot (n-2) \cdot (n-1) \cdot n$ .

## 2. Popis problému

Při určení poslední nenulové číslice faktoriálu přirozeného čísla narážíme na několik problémů při násobení:

- 1) při násobení sudých čísel s číslem končícím pětkou nebo součin jakéhokoliv čísla s číslem končícím nulou dochází k nárůstu nul na konci výsledného čísla (například:  $2 \cdot 5 = 10$ ,  $35 \cdot 6 = 210$ ,  $123 \cdot 100 = 12300$  atd.)
- 2) dalším problémem je to, že při násobení velmi rychle dochází k nárůstu řádu výsledných čísel, jelikož řešíme problém pomocí výpočetní techniky a máme k dispozici pouze rozsah 32bitů což je  $2^{32} = 4,294,967,296$  už při čísle  $13!$  což je 6,227,020,800 bychom překročili stanovený rozsah musíme vymyslet řešení, jak tento problém obejít.

## 3. Řešení problému

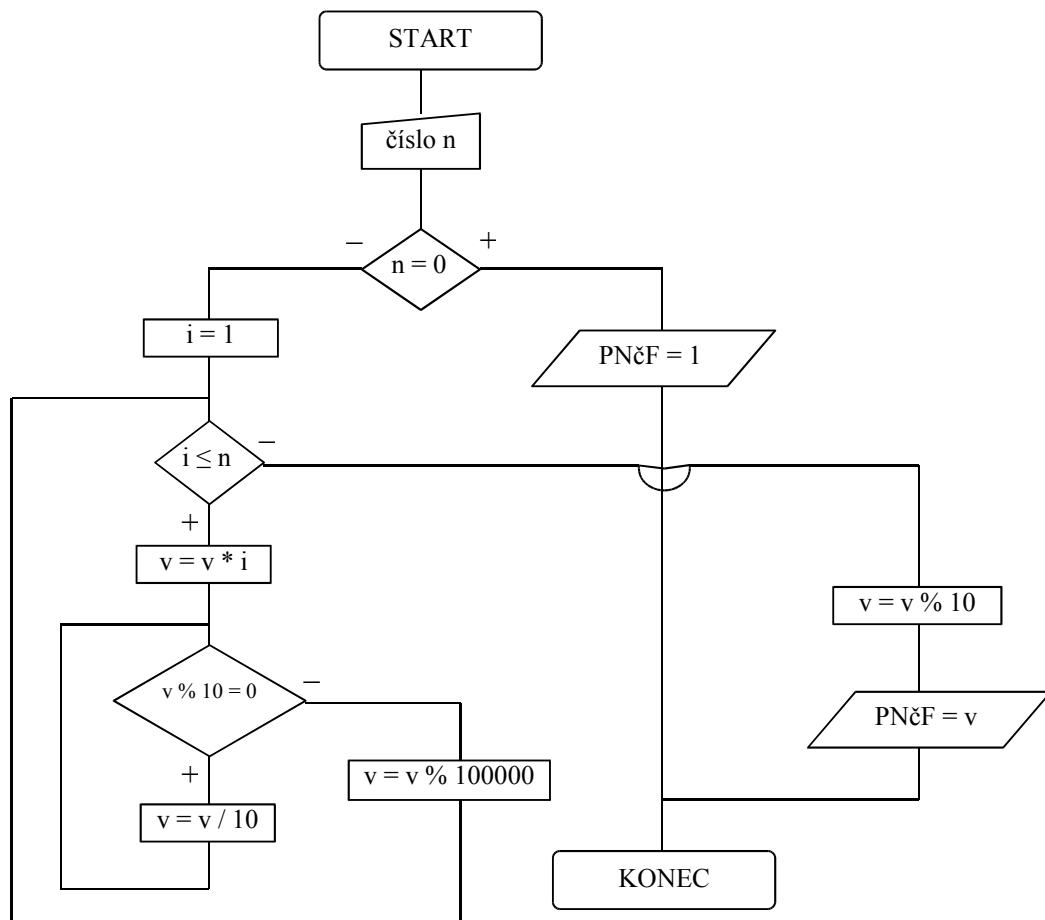
ad 1) odstranění (pro nás) přebytečných nul je celkem snadné. Koncové nuly můžeme „zanedbat“ a to kdykoliv během výpočtu, protože  $0 \cdot x = 0$  (důkaz viz. skripta: Diskrétní Matematika, 2004, doc. Petr Hliněný, str. 19, Příklad 3.1.) stejně jako  $abc0 \cdot x = abc \cdot x0$  a počet nul na konci se nemění.

**Řešení v oblasti výpočetní techniky:** jednoduchý algoritmus spočívá v jednoduchém cyklu, kdy stačí číslo vydělit číslem 10 tolikrát, kolik je počet nul na konci čísla.

ad 2) co se týče druhého problému, jako řešení lze použít násobení jen části čísla. Konkrétně pak součin nejnižších řádů jednotlivých čísel. Jelikož je našim úkolem zjistit poslední nenulovou číslici, nepotřebujeme vysoké řády čísla.

**Řešení v oblasti výpočetní techniky:** jednoduchý algoritmus spočívá v dělení modulo (zbytek po celočíselném dělení) číslem  $10^n$  kde  $n$  je počet platných číslic zprava. Pro naše účely je dostačující 5 platných číslic zprava. Alternativní řešení ve Visual Basicu dovoluje použít funkci Right(), která má dva parametry Right(*řetězec*, *délka*) – funkce vybere počet číslic zprava určených *délkou* z řetězce jménem *řetězec*. Pozn.: přetypování z číselného datového typu na datový typ String probíhá automaticky.

## 4. Vývojový diagram



## 5. Popis algoritmu

Podmínka – je číslo  $n = 0$ ?

Jestliže **ano**, pak poslední nenulová číslice faktoriálu (dále jen **PNČF**) je rovna **1**.

**Konec algoritmu.**

Jestliže **ne**, pak:

**Cyklus – od 1 do  $n$ .** Cykluj tak dlouho dokud je  $i$  menší nebo rovno číslu  $n$ .

Výpočet –  $v = v * i$ . Součin předchozího výsledku s proměnnou cyklu  $i$ .

**Podmíněný cyklus** – Jestliže je na konci nula ( $v \bmod 10 = 0$ ), pak tu nulu odstraň  $v = v / 10$  a pokračuj tak dlouho dokud není na konci nenulová číslice, tzn. tak dlouho dokud je na konci nula.

Výpočet –  $v = v \% 100000$ . Do výsledku vlož pouze posledních 5 platných číslic.

**Konec cyklu od 1 do  $n$ .**

Výpočet –  $v = v \% 10$ . Výsledek je roven poslednímu číslu.

**PNČF =  $v$ .**

**Konec algoritmu.**

## 6. Implementace

```
Sub Visual Basic()  
  
Dim n As Long 'přirozené číslo typu 32bit integer  
Dim v As Long 'proměnná pro mezivýsledky typu 32bit integer  
v = 1 'počáteční hodnota  
  
If n = 0 Then 'podmínka - v případě, že n = 0 výsledkem je 1  
    v = 1  
Else  
    For i = 1 To n 'cyklus od 1 po hodnotu n  
        v = v * i 'součin mezivýsledku s proměnnou i  
        Do While (v Mod 10 = 0)  
            'alternativa (Right(v, 1) = 0) 'podmíněný cyklus  
            v = v / 10 'odstraňuje nuly  
            ' * na tomto místě by mohlo být počítadlo nul  
        Loop  
        v = v Mod 100000 'posledních 5 platných číslic  
        'alternativa v = Right(v, 5)  
    Next i  
    v = v Mod 10 'alternativa v = Right(v, 1) - poslední číslice  
  
End If  
End Sub
```

```
void main() { //C++  
  
int n, v; //čísla typu integer  
v = 1; //počáteční hodnota  
  
if (n == 0) //podmínka - v případě, že n = 0 výsledkem je 1  
    v = 1;  
else {  
    for (int i = 1; i <= n; i++) { //cyklus od 1 do n  
        v = v * i; //součin mezivýsledku s proměnnou i  
        while ((v % 10) == 0) { //podmíněný cyklus  
            v = v / 10; //odstraňuje nuly  
            // * na tomto místě by mohlo být počítadlo nul  
        } //end while  
        v = v % 100000; //posledních 5 platných číslic  
    } //end for  
    v = v % 10; //poslední číslice  
  
} //end else  
  
} //end void main()
```

## 7. Závěr

Algoritmus je teoreticky navržen pro výpočet hodnot do  $n = 42\,950$ . Což je  $2^{32} / 99999$ .  $2^{32}$  – 32bitové číslo děleno největším možným číslem vyjádřeným 5 platnými číslicemi. Prakticky toto číslo není ověřeno z důvodu velmi dlouhého výpočtu. Nejvyšší mnou ověřená hodnota pomocí programu ve VB je pro  $n = 7000!$  kde hodnota PNČF čísla  $7000! = 4$ .

V případě, že by nás zajímal počet nul před PNČF, je zřejmé, že nejjednodušší způsob je umístění „počítadla“ v místě \*, kde se nuly odstraňují.

Výsledky PNČF pro **2004! = 2** a pro **2005! = 6**.

Několik náhodně vybraných výsledků:

0028! = 8	0558! = 4	1004! = 2	1521! = 2
0031! = 2	0568! = 8	1027! = 2	1534! = 4
0045! = 2	0579! = 8	1049! = 2	1549! = 2
0090! = 2	0579! = 8	1066! = 4	1570! = 6
0091! = 6	0591! = 8	1067! = 2	1580! = 2
0107! = 4	0596! = 4	1086! = 8	1597! = 2
0111! = 2	0601! = 8	1122! = 8	1628! = 8
0112! = 6	0603! = 8	1150! = 2	1649! = 8
0121! = 8	0652! = 6	1159! = 2	1659! = 2
0161! = 2	0706! = 6	1166! = 2	1669! = 2
0195! = 2	0728! = 4	1178! = 4	1725! = 4
0200! = 4	0747! = 4	1184! = 6	1742! = 4
0206! = 6	0757! = 2	1245! = 6	1811! = 6
0212! = 4	0764! = 8	1255! = 2	1821! = 2
0320! = 4	0780! = 2	1263! = 8	1827! = 6
0325! = 8	0802! = 6	1266! = 6	1832! = 4
0372! = 4	0809! = 8	1293! = 2	1838! = 8
0415! = 4	0820! = 8	1295! = 2	1897! = 4
0453! = 6	0825! = 6	1352! = 4	1899! = 6
0487! = 8	0828! = 4	1355! = 4	1923! = 4
0487! = 6	0856! = 8	1388! = 4	1958! = 8
0522! = 2	0860! = 4	1390! = 8	1959! = 8
0527! = 4	0915! = 2	1411! = 2	1960! = 6
0539! = 6	0925! = 2	1418! = 6	1972! = 8
0556! = 6	0937! = 6	1425! = 2	1998! = 4