

# Projekt 2 - Iterační výpočty

## Popis projektu

---

Implementujte výpočet přirozeného logaritmu a exponenciální funkce s obecným základem pouze pomocí matematických operací  $+$ ,  $-$ ,  $*$ ,  $/$ .

## Detailní specifikace

---

### Překlad a odevzdání zdrojového souboru

Odevzdání: Program implementujte ve zdrojovém souboru `proj2.c`. Zdrojový soubor odevzdejte prostřednictvím informačního systému.

Překlad: Program překládejte s následujícími argumenty:

```
$ gcc -std=c99 -Wall -Wextra -Werror proj2.c -lm -o proj2
```

### Syntax spuštění

Program se spouští v následující podobě:

```
./proj2 --log X N
```

nebo

```
./proj2 --pow X Y N
```

Argumenty programu:

- `--log X N` požadavek pro výpočet přirozeného logaritmu z čísla  $X$  v  $N$  iteracích (Taylorova polynomu a zřetěženého zlomku).
- `--pow X Y N` požadavek pro výpočet exponenciální funkce z čísla  $Y$  s obecným základem  $X$  v  $N$  iteracích (Taylorova polynomu a zřetěženého zlomku)

### Implementační detaily

Je zakázané použít funkce z matematické knihovny. Jedinou výjimkou jsou funkce `log` a `pow` použité pouze pro srovnání výpočtů, funkce `fabs` (kvůli své trivialitě), funkce `isnan` a `isinf` a konstanty `NAN` a `INFINITY`. Ve všech výpočtech používejte typ `double`.

## Implementace logaritmu

Funkci logaritmu implementujte dvakrát a to pomocí Taylorova polynomu a zřetězených zlomků.

### 1. podúkol - Implementace Taylorova polynomu

Logaritmus pomocí Taylorova polynomu implementujte ve funkci s prototypem:

```
double taylor_log(double x, unsigned int n);
```

kde n udává rozvoj polynomu (počet členů). Taylorův polynom pro funkci logaritmu implementujte podle vzorce:

$$\log(1 - x) = -x - \frac{x^2}{2} - \frac{x^3}{3} - \frac{x^4}{4} - \dots$$

pro  $0 < x < 2$  a

$$\log(x) = \sum_{n=1}^{\infty} \frac{((x - 1)/x)^n}{n}$$

pro  $x > 1/2$ . Doporučená mezní hodnota mezi těmito dvěma polynomy je 1.

### 2. podúkol - Implementace zřetězeného zlomku

Logaritmus pomocí zřetězených zlomků (viz demonstrační cvičení) implementujte ve funkci s prototypem:

```
double cfrac_log(double x, unsigned int n);
```

kde n udává rozvoj zřetězeného zlomku. Funkci implementujte podle vzorce:

$$\log\left(\frac{1+z}{1-z}\right) = \frac{2z}{1 - \frac{z^2}{3 - \frac{4z^2}{5 - \frac{9z^2}{7 - \frac{16z^2}{9 - \frac{25z^2}{11 - \frac{36z^2}{13 - \dots}}}}}}}$$

### 3. podúkol - Výpočet exponenciální funkce s obecným základem

Exponenciální funkci s obecným základem počítejte ve funkci s prototypem:

```
double taylor_pow(double x, double y, unsigned int n);
```

a

```
double taylorcf_pow(double x, double y, unsigned int n);
```

kde  $n$  udává rozvoj polynomu (počet členů) a parametry  $x$  a  $y$  odpovídají parametrům funkce `pow` z matematické knihovny. Taylorův polynom pro exponenciální funkci implementujte podle vzorce:

$$a^x = e^{x \ln a} = 1 + \frac{x \ln a}{1!} + \frac{x^2 \ln^2 a}{2!} + \frac{x^3 \ln^3 a}{3!} + \dots$$

pro  $a > 0$ .

Pro výpočet přirozeného logaritmu použijte funkci `taylor_log` v případě funkce `taylor_pow` a funkci `cfrac_log` v případě funkce `taylorcf_pow`.

#### Výstup programu

V případě výpočtu logaritmu (argument `--log`) program tiskne následující řádky:

```
log(X) = LOG_X
cfrac_log(X) = CFRAC_LOG_X
taylor_log(X) = TAYLOR_LOG_X
```

V případě výpočtu exponenciální funkce (argument `--pow`) program tiskne následující řádky:

```
pow(X,Y) = POW
taylor_pow(X,Y) = TAYLOR_POW
taylorcf_pow(X,Y) = TAYLORCF_POW
```

kde:

- $x$  a  $y$  jsou hodnoty zadané argumentem příkazové řádky (odpovídají formátu `printf %g`),
- `LOG_X` je hodnota logaritmu z matematické knihovny,
- `CFRAC_LOG_` jsou hodnoty logaritmu vypočteného pomocí zřetěženého zlomku,
- `TAYLOR_LOG_` jsou hodnoty logaritmu vypočteného pomocí Taylorova polynomu,
- `POW` je hodnota exponenciální funkce z matematické knihovny,
- `TAYLOR_POW` je hodnota vypočtená pomocí funkce `taylor_pow`,
- `TAYLORCF_POW` je hodnota vypočtená pomocí funkce `taylorcf_pow`,
- všechny `*LOG_*` a `*POW` hodnoty odpovídají formátu `%.12g`.

### Příklady vstupů a výstupů

*Číselné údaje nemusí přesně odpovídat vaší implementaci. Výsledek závisí na způsobu implementace a optimalizaci.*

```
$ ./proj2 --log 1.131401114526 4
    log(1.1314) = 0.123456789012
  cfrac_log(1.1314) = 0.123456789012
taylor_log(1.1314) = 0.123452108537
$ ./proj2 --pow 1.23 4.2 4
    pow(1.23,4.2) = 2.38562110403
  taylor_pow(1.23,4.2) = 2.38026034593
taylorcf_pow(1.23,4.2) = 2.38079698151
```

## Hodnocení

---

Na výsledném hodnocení mají hlavní vliv následující faktory:

- implementace algoritmických schemat pro iterační výpočty,
- výpočet logaritmu a exponenciální funkce,
- ošetření neočekávaných stavů.

## Prémie

Prémiové body (max 4) je možné získat implementací alternativní funkce k funkcím log a pow. Alternativní funkce budou implementovány v prototypch:

```
double mylog(double x);
```

a

```
double mypow(double x, double y);
```

Obě funkce budou podle hodnoty zadaného argumentu volit nejpřesnější typ výpočtu (Taylorův polynom nebo zřetěžené zlomky) a minimální počet iterací pro požadovanou přesnost. Nechtě požadovaná přesnost je na 8 významných číslic (tj. odpovídá výstupnímu formátu %.7e přesného výsledku). Podmínkou pro udělení prémiových bodů je úspěšná obhajoba projektu a prémiového vypracování.