

FLP 2022/2023 – logický projekt: Prolog

Toto je zadání logického projektu do předmětu Funkcionální a logické programování 2022/2023. Za projekt zodpovídá **Ing. Radek Hranický, Ph.D.** Konzultace jsou možné kdykoliv po předchozí domluvě; případně elektronickou poštou (hranicky@fit.vut.cz) či prostřednictvím diskusního fóra předmětu, které najdete v IS VUT v sekci e-Learning.

Za vypracovaný projekt lze získat až 8 bodů. Minimální počet bodů pro získání zápočtu jsou 3. **Hodnocena bude míra splnění zadání, kvalita řešení, čistota a kvalita kódu a vhodné užití komentářů.** Za inovativní přístup či obzvláště kvalitní řešení lze získat prémiové body navíc.

Obecné pokyny pro všechna zadání jsou sepsány ve zvláštním dokumentu, nezapomeňte na jejich dodržování. **Projekt je nutné vypracovat samostatně** – duplikáty jsou kontrolovány i strojově a pokus o opsání nebo přílišnou “inspiraci” cizím kódem je velmi dobře poznat! Případné nejasnosti konzultujte včas, na námítky po odevzdání nebude brán zřetel. U všech zadání je kladen důraz především na správné chování v případě správného vstupu. Cílem projektu je vyzkoušet si principy logického programování, nikoliv dokonalé ošetření vstupu. Obecně také platí, že pokud existuje cesta k vyřešení problému, Váš program ji musí nalézt. Pokud existuje více cest, může Váš program zvolit libovolnou z nich.

Úvod všech zdrojových textů musí obsahovat *název projektu, login a jméno autora*. Součástí řešení je Makefile (projekty budou překládány pomocí make), cílový program pojmenujte **f1p22-log**. Makefile ani přeložený program **neumísťujte do podadresářů**.¹ Projekt vypracujte v prostředí SWI-Prolog, za referenční stroj je považován server merlin. **Pokud nebude projekt funkční na serveru merlin, nebudou uděleny body. Před odevzdáním vždy vyzkoušejte spuštění projektu na serveru merlin!**

Spolu s projektem odevzdejte i několik vstupních souborů, na nichž je možné váš projekt otestovat, do dokumentace uveďte pro každý takový soubor přibližnou dobu výpočtu. Prosím neodevzdávejte zbytečné soubory.

Následuje popis jednotlivých variant zadání.

¹ Tzn. např. sekvencí příkazů

unzip f1p-log-xlogin00.zip

make

./f1p22-log < /cesta_k_testum/vstup1.txt > /cesta_k_vysledkum/vystup1.txt

dojde k rozbalení odevzdaného archivu, zkompilování projektu a spuštění na testovacích datech.

Kostra grafu

Vaším úkolem je napsat program schopný nalézt všechny kostry zadaného neorientovaného grafu. Kostra je takový podgraf, který je tvořen všemi vrcholy původního grafu a zároveň je to strom (je souvislý a neobsahuje žádnou kružnici). Na vstupu budete mít zadány neorientované hrany grafu ve formě dvojic oddělených mezerou. Tato písmena značí vrcholy grafu. Pokud by vstupní graf nebyl souvislý, tak se na výstup nevytiskne nic.

Vrcholy jsou vždy označeny pomocí velkých písmen anglické abecedy (A až Z). Jednotlivé řádky vstupního souboru mají tvar „<V1> <V2>“ (bez uvozovek). Jakýkoliv jiný řádek je ignorován. Jako výstup programu se očekává seznam koster, kde na jednom řádku je jedna kostra. Každá kostra má výstup ve tvaru „<V1>-<V2> <V3>-<V4> ... <Vn-1>-<Vn>“. Každá dvojice vrcholů reprezentuje hranu dané kostry. Pořadí vrcholů v hraně, jednotlivých hran, nebo pořadí koster ve výstupu je libovolné, každá kostra však musí být unikátní.

Ukázka vstupu a výstupu:

```
A B
A C
A D
B C
C D
```

```
A-B A-C A-D
A-B A-C C-D
A-B A-D B-C
A-B A-D C-D
A-B B-C C-D
A-C A-D B-C
A-C B-C C-D
A-D B-C C-D
```