Projekt 2 - Práce s datovými strukturami

Na projektu lze pracovat v týmu 3-5 členů. V týdnu 1.-5.11. se seznamte se zadáním projektu a zformujte si tým, ve kterém chcete na projektu pracovat. Členové týmu musí patřit do stejné skupiny laboratorního cvičení.

Motivace projektu

V prvním projektu jsme si vyzkoušeli jednoduché práce s textovými řetězci a základními cykly. V tomto projektu spojíme znalosti programování a diskrétní matematiky. Projekt zahrnuje zpracování vstupu, dynamickou alokaci a algoritmy pro průchody v datových strukturách.

Popis projektu

Cílem projektu je vytvořit program, který bude implementovat základní matematické operace nad množinami a binárními relacemi. Vstupem programu budou textová data reprezentující množiny a relace a zadání operací. Výsledek zpracování bude program tisknout na standardní výstup.

Detailní specifikace

Program implementujte ve zdrojovém souboru *setcal.c* (Set Calculator). Vstupní data budou čtena ze souboru, jehož jméno bude zadáno argumentem příkazové řádky. Program provádí operace zadané v souboru po řádcích jedním průchodem. Výsledek zpracování každého řádku program tiskne na standardní výstup (tedy počet řádků na výstupu odpovídá počtu řádků vstupního souboru).

Překlad a odevzdání zdrojového souboru

Odevzdání: Odevzdejte zdrojový soubor *setcal.c* prostřednictvím informačního systému.

Překlad: Program překládejte s následujícími argumenty

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

Syntax spuštění

Program se spouští v následující podobě: (./setcal značí umístění a název programu):

```
./setcal FILE
```

Formát vstupního souboru

Textový soubor se skládá ze tří po sobě následujících částí:

1. Definice univerza - jeden řádek začínající "U " a pokračující mezerou oddělenými prvky,
2. Definice množin a relací - jeden nebo více řádků začínající "S " nebo "R " a pokračující mezerou oddělenými prvky (řádek začínající "S" indikuje definici množiny, "R" slouží pro definici relace),
3. Příkazy nad množinami a relacemi - jeden nebo více řádků začínající "C " a pokračující identifikátorem příkazu.

Univerzum

Prvky univerza ~~mohou být~~jsou řetězce obsahující malá a velká písmena anglické abecedy. Délka řetězce je maximálně 30 znaků. Prvky univerza nesmí obsahovat identifikátory příkazů (viz níže) a klíčová slova true a false. Všechny prvky v množinách a relacích musí patřit do univerza. Příklad:

```
U Apple Lemon Orange Banana Peach
```

Množiny

Každá množina je definovaná na jednom řádku mezerou oddělenými prvky z univerza. Identifikátorem množiny je číslo řádku, na kterém je množina definovaná (vzhledem k tomu, že na prvním řádku souboru je univerzum, identifikátory množin tedy začínají číslem 2). Identifikátory množin jsou použity v operacích (viz níže). Příklad definice množiny:

```
S Apple Banana Peach
```

Relace

Každá relace je definována výčtem dvojic. Dvojice je ohraničená kulatými závorkami, první a druhý prvek dvojice jsou oddělené mezerou. Jednotlivé dvojice jsou oddělené mezerou. Příklad:

```
R (Apple Banana) (Apple Peach) (Apple Apple)
```

Příkazy

Každý příkaz je definován na jednom řádku, začíná identifikátorem příkazu a argumenty příkazu jsou oddělené mezerou (od identifikátoru i mezi sebou). Argumenty příkazu jsou číselné identifikátory množin a relací (celá kladná čísla, číslo 1 identifikuje množinu univerza). Příklad:

```
C minus 1 2
```

Příkazy nad množinami

Příkaz pracuje nad množinami a jeho výsledkem je buď množina (v tom případě tiskne množinu ve stejném formátu jako se očekává ve vstupním souboru, tj. začíná "S " a pokračuje mezerou oddělenými prvky) nebo je výsledkem pravdivostní hodnota (v tom případě tiskne true nebo false na samostatný řádek) nebo je výsledkem přirozené číslo (které se tiskne na samostatný řádek).

- `empty A` - tiskne true nebo false podle toho, jestli je množina definovaná na řádku A prázdná nebo neprázdná.
- `card A` - tiskne počet prvků v množině A (definované na řádku A).
- `complement A` - tiskne doplněk množiny A.
- `union A B` - tiskne sjednocení množin A a B.
- `intersect A B` - tiskne průnik množin A a B.

- `minus A B` - tiskne rozdíl množin A \ B.
- `subseteq A B` - tiskne true nebo false podle toho, jestli je množina A podmnožinou množiny B.
- `subset A B` - tiskne true nebo false, jestli je množina A vlastní podmnožina množiny B.
- `equals A B` - tiskne true nebo false, jestli jsou množiny rovny.

Příkazy nad relacemi

Příkaz pracuje nad relacemi a jeho výsledkem je buď pravdivostní hodnota (tiskne true nebo false), nebo množina (tiskne množinu ve formátu jako ve vstupnímu souboru).

- `reflexive R` - tiskne true nebo false, jestli je relace reflexivní.
- `symmetric R` - tiskne true nebo false, jestli je relace symetrická.
- `antisymmetric R` - tiskne true nebo false, jestli je relace antisymetrická.
- `transitive R` - tiskne true nebo false, jestli je relace tranzitivní.
- `function R` - tiskne true nebo false, jestli je relace R funkcí.
- `domain R` - tiskne definiční obor funkce R (lze aplikovat i na relace - první prvky dvojic).
- `codomain R` - tiskne obor hodnot funkce R (lze aplikovat i na relace - druhé prvky dvojic).
- `injective R A B` - tiskne true nebo false, jestli je funkce R injektivní. A a B jsou množiny; a∈A, b∈B, (a,b)∈R.
- `surjective R A B` - tiskne true nebo false, jestli je funkce R surjektivní. A a B jsou množiny; a∈A, b∈B, (a,b)∈R.
- `bijective R A B` - tiskne true nebo false, jestli je funkce R bijektivní. A a B jsou množiny; a∈A, b∈B, (a,b)∈R.

Implementační detaily

- Maximální podporovaný počet řádků je 1000.
- Na pořadí prvků v množině a v relaci na výstupu nezáleží.
- Všechny prvky množin a v relacích musí patřit do univerza. Pokud se prvek v množině nebo dvojice v relaci opakuje, jedná se o chybu.

Příklady použití

```
$ cat sets.txt
U a b c x y z
S a b c x
S x y z
C intersect 2 3
C minus 2 3
```

```
$ ./setcal sets.txt
U a b c x y z
S a b c x
S x y z
S x
S a b c
```

```
$ cat rel.txt
U dad mom girl boy man woman
R (dad boy) (dad girl) (mom boy) (mom girl)
R (dad man) (boy man) (mom woman) (girl woman)
C codomain 2
C function 3
```

```
$ ./setcal rel.txt
U dad mom girl boy man woman
R (dad boy) (dad girl) (mom boy) (mom girl)
R (dad man) (boy man) (mom woman) (girl woman)
S boy girl
true
```

Tipy pro implementaci

- Vytvořte datový typ pro množinu. Uvažujte předem neznámý počet prvků. Každý prvek (řetězec) by měl být dynamicky alokován.
- Implementujte funkci pro tisk množiny.
- Implementujte funkci pro načtení množiny ze souboru do paměti.
- Vytvořte základní (unární operace) nad množinou.
- Vytvořte datový typ pro načtení více množin. Implementujte další operace nad množinami.
- Vytvořte datový typ pro relaci.
- Vytvořte funkci pro načtení relace ze souboru do paměti.
- Implementujte operace nad relacemi.

Neočekávané chování

Na chyby za běhu programu reagujte obvyklým způsobem: Na neočekávaná vstupní data, formát vstupních dat nebo chyby při zpracování operací reagujte přerušením programu se stručným a výstižným chybovým hlášením na příslušný výstup a odpovídajícím návratovým kódem. Hlášení budou v kódování ASCII česky nebo anglicky. Všechna alokovaná paměť musí být uvolněna a otevřené soubory uzavřené.

Hodnocení

Na výsledném hodnocení mají hlavní vliv následující faktory:

1. přeložitelnost zdrojového souboru,
    1. zdrojový soubor musí být přeložitelný a běžet na GNU/Linux, například na serveru merlin.fit.vutbr.cz (pozn. Kód má být napsán v jazyku C podle standardu ISO C 99, který má fungovat stejně na všech platformách. Jestliže kód nebude fungovat na stroji merlin, nebude fungovat na spoustě dalších platformách).
2. dekompozice problému na podproblémy (vhodné funkce, vhodná délka funkcí a parametry funkcí),
3. správná volba datových typů, tvorba nových typů,
4. správná funkcionalita (bodové hodnocení rostoucí v následujícím pořadí):
    1. s možným omezením na velikost množin nebo jejich počet (bez dynamické alokace)
    2. neomezené množiny a relace
    3. podpora pouze množinových operací
    4. podpora relací nad množinami
5. ošetření chybových stavů.

Prémiové hodnocení

Prémiové hodnocení je dobrovolné a lze za něj získat bonusové body. Podmínkou pro udělení prémiových bodů je úspěšná obhajoba projektu a výborné vypracování standardního zadání. Výsledné hodnocení je plně v kompetenci vyučujícího, kterému bude projekt obhajován a který bude projekt hodnotit. Výše

prémiových bodů závisí také na sofistikovanosti řešení. Prémiová implementace by měla zahrnovat pokročilé příkazy.

Pokročilé příkazy

Pokud je výsledkem příkazu relace, tiskne příkaz relaci ve stejném formátu jako ve vstupním souboru (začíná "R ").

- `closure_ref R` - tiskne reflexivní uzávěr relace R
- `closure_sym R` - tiskne symetrický uzávěr relace R
- `closure_trans R` - tiskne tranzitivní uzávěr relace R
- Rozšíření všech příkazů, jejichž výsledkem je množina nebo relace, definuje novou množinu nebo relaci identifikovanou číslem řádku, na kterém se nachází daná operace.
- Rozšíření všech příkazů, které tisknou true nebo false o další argument N. V případě, že operace končí s výsledkem false, následující řádek, který se zpracovává, bude na řádku N (nikoliv bezprostředně následující).
- `select A N` - vybere náhodný prvek z množiny nebo relace A a tiskne ho. V případě, že je množina A prázdná, přeskočí vykonávání příkazu na řádek N vstupního souboru. N v takovém případě musí označovat existující řádek ve vstupním souboru.

Id stránky: 156, zobrazena: 125874, verze: 18018, dne: 2021-12-08 15:14:47 uložil: smrcka Nahoru