Projekt 1 - Práce s textem

Motivační příklad

Představme si virtuální klávesnici v navigaci na palubní desce auta. Navigace má celou databázi adres a očekává od uživatele vstup pomocí virtuální klávesnice. Aby se vstup uživateli usnadnil, jsou na klávesnici v daném okamžiku povoleny pouze vybrané klávesy – ty klávesy, při jejichž použití by vyhledáváný řetězec stále vedl ke známé adrese. Například navigace zná města Brno, Beroun a Bruntál, uživatel již zadal písmena "BR" a na navigaci budou tedy povoleny pouze klávesy "N" (vedoucí k "Brno") a "U" (vedoucí k "Bruntál").

Popis projektu

Cílem projektu je vytvořit program, který by emuloval algoritmus výpočtu povolení a zakázání kláves na virtuální klávesnici navigace. Vstupem programu jsou data známých adres a uživatelem zadaný řetězec. Výstup programu bude obsahovat znaky, které mají být na klávesnici povoleny.

Detailní specifikace

Program implementujte v jednom zdrojovém souboru *keyfilter.c*. Vstupní data budou čtena ze standardního vstupu (stdin), výstup bude tisknut na standardní výstup (stdout).

Překlad a odevzdání zdrojového souboru

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

Překlad: Program překládejte s následujícími argumenty

```
$ gcc -std=c11 -Wall -Wextra -Werror keyfilter.c -o keyfilter
```

Syntax spuštění

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

```
./keyfilter ADRESA
```

Pokud je program spuštěn bez argumentů, bere zadanou adresu jako prázdný řetězec.

Vstupní databáze adres

Databáze adres jsou textová data, u kterých každý řádek označuje jednu adresu. Každý řádek obsahuje maximálně 100 znaků. Seznam adres je neuspořádaný. U všech dat nezáleží na velikosti písmen (tzv. case insensitive). Program musí podporovat alespoň 42 adres.

Výstup programu

Výstup programu může být trojího druhu:

1. adresa nalezena,
2. adresa vyžaduje specifikaci,
3. adresa nenalezena.

1. Adresa nalezena

Found: S

Tento výstup se tiskne, pokud je v databázi adres nalezena jediná adresa S, jejíž prefix odpovídá uživatelem zadané adrese ADRESA. (Pozn. prefix P řetězce S je takový řetězec, u kterého řetězec S začíná řetězcem P).

2. Adresa vyžaduje specifikaci

Enable: CHARS

Pokud je v databázi adres nalezeno více adres odpovídající danému prefixu ADRESA, program pomocí takto naformátovaného řádku vytiskne seznam povolených kláves CHARS. CHARS je abecedně seřazený seznam znaků, u nichž pro každý znak C platí, že v databázi adres existuje adresa, jejíž prefix odpovídá spojení řetězce ADRESA s daným znakem C.

3. Adresa nenalezena

Not found

Pokud v databázi adres neexistuje adresa, jejíž prefix by odpovídal zadanému řetězci ADRESA, vytiskne program toto hlášení.

Omezení v projektu

V projektu je povoleno použít všechny funkce standardu C kromě níže uvedených:

- volání z rodiny malloc a free – práce s dynamickou pamětí není v tomto projektu zapotřebí,
- volání fopen, fclose, fscanf, feof, fseek, ftell – práce s externími soubory není v tomto projektu zapotřebí (vystačíte si se stdin, stdout a stderr),
- volání qsort, lsearch, bsearch a hsearch – cílem je zamyslet se nad algoritmizací a strukturou dat, není ani potřeba implementovat vlastní řadicí algoritmus.

Neočekávané chování

Na chyby za běhu programu reagujte obvyklým způsobem: Na neočekávaná vstupní data, formát vstupních dat nebo chyby při volání funkcí reagujte přerušením programu se stručným a výstižným chybovým hlášením na příslušný výstup a odpovídajícím návratovým kódem. Hlášení budou v kódování ASCII česky nebo anglicky.

Příklady vstupů a výstupů

Pomocný soubor adres:

$ cat adresy.txt

Praha

Brno

Bruntal

Bratislava

Příklad hledání slova brno

$ ./keyfilter <adresy.txt

Enable: BP

$ ./keyfilter b <adresy.txt

Enable: R

$ ./keyfilter br <adresy.txt

Enable: ANU

$ ./keyfilter brn <adresy.txt

Found: BRNO

$ ./keyfilter be <adresy.txt

Not found

Hodnocení

Na výsledném hodnocení mají hlavní vliv následující faktory:

- přeložitelnost zdrojového souboru,
- formát zdrojového souboru (členění, zarovnání, komentáře, vhodně zvolené identifikátory),
- dekompozice problému na podproblémy (vhodné funkce, vhodná délka funkcí a parametry funkcí),
- správná volba datových typů, případně tvorba nových typů,
- správná funkcionalita vyhledání a
- ošetření chybových stavů.

Priority funkcionality

1. Vyhledání prefixů v řetězci a odpovídajících adres.
2. Datové struktury pro povolené klávesy a vyhledání kláves.
3. Proudové zpracování vstupních adres a tisk výsledku hledání.

Prémiové hodnocení

V případě implementace bez omezení počtu podporovaných adres (implementace proudového zpracování) je možné získat 1-2 prémiové body. Získání prémiových bodů je podmíněno správnou implementací a dodržení zadání.