

Základy programování (IZP 22/23Z)

Zadání projektu 1 - Práce s textem

Projekt 1 - Práce s textem

Motivační příklad

Snad všichni známe telefonní seznam v chytrých telefonech. Vyhledávání tam většinou funguje i za pomoci numerické klávesnice. Uživatel zadává posloupnosti číslic, ale každým stiskem klávesy myslí jeden ze znaků, které daná číslice reprezentuje (například číslo 5 reprezentuje samotnou 5 či jedno z písmen j, k nebo l). Telefon pak filtruje telefonní seznam podle dané posloupnosti, přičemž se snaží v seznamu najít libovolný kontakt, jehož jméno nebo telefonní číslo obsahuje sekvenci (nepřerušenou nebo rozdělenou) zadaných znaků.

Popis projektu

Cílem projektu je vytvořit program, který by emuloval zjednodušený algoritmus hledání kontaktů zadaných pomocí posloupnosti číslic. Vstupem programu je posloupnost číslic. Výstup programu bude obsahovat seznam kontaktů, které odpovídají zadanému číselnému filtru.

Detailní specifikace

Program implementujte ve zdrojovém souboru "t9search.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 "t9search.c" prostřednictvím informačního systému.

Překlad: Program překládejte s následujícími argumenty:

```
gcc -std=c99 -Wall -Wextra -Werror t9search.c -o t9search
```

Syntax spuštění

Program se spouští v následující podobě: (./t9search značí umístění a název programu):

```
./t9search CISLO
```

CISLO zde představuje hledané kritérium, sekvenci číslic 0 až 9, které reprezentují sekvenci hledaných znaků.

Pokud je program spuštěn bez argumentů, program se bude chovat, jako by hledanému kritériu odpovídal každý kontakt ze seznamu.

Implementační detaily

Vstupní telefonní seznam

Telefonní seznam jsou ASCII textová data. Každý záznam o kontaktu obsahuje dva **neprázdné (bylo přidáno 2022-10-17, není nutné toto akceptovat)** řádky: jméno osoby a jeho telefonní číslo. Každý řádek obsahuje maximálně 100 znaků, jinak se jedná o neplatná data. Seznam kontaktů je neuspořádaný. U všech dat nezáleží na velikosti písmen (tzv. case insensitive). Program musí podporovat alespoň 42 kontaktů.

Kritérium nalezení kontaktu

Pro zjednodušení budeme v tomto projektu uvažovat hledání "nepřerušené posloupnosti" zadaných znaků/číslic v telefonním kontaktu. Nepřerušenou posloupností znaků se myslí takový podřetězec, u kterého se na každé pozici vyskytuje znak, který odpovídá jednomu ze znaků, které reprezentuje číslice na stejné pozici v zadaném řetězci. Každá číslice kromě sebe sama reprezentuje ještě následující znaky: 2 (abc), 3 (def), 4 (ghi), 5 (jkl), 6 (mno), 7 (pqrs), 8 (tuv), 9 (wxyz), 0 (+).

Výstup programu

Výstup programu může být dvojího druhu:

- žádný kontakt nenalezen,
- nalezen jeden a více kontaktů.

Kontakt nenalezen

Not found

Pokud v telefonním seznamu neexistuje kontakt, u něhož by nějaká část odpovídala hledanému řetězci CISLO, vytiskne program toto hlášení.

Kontakt(y) nalezen(y)

Každý kontakt, který odpovídá hledanému kritériu, program vytiskne na jeden řádek ve formátu "JMENO, TELCISLO", kde JMENO je jméno kontaktu z telefonního seznamu a TELCISLO je záznam o telefonním čísle. Program tiskne všechny kontakty, které kritériu odpovídají. Na pořadí řádků nezáleží. Stejně tak nezáleží ani na velikosti písmen.

Omezení v projektu

Je zakázané použít následující funkce:

- volání z rodiny malloc a free - práce s dynamickou pamětí není v tomto projektu zapotřebí,
- volání z rodiny fopen, fclose, fscanf, ... - práce se soubory (dočasnými) není v tomto projektu žádoucí,
- volání qsort, lsearch, bsearch a hsearch - cílem je zamyslet se nad algoritmizací a strukturou dat.

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 (slovensky) nebo anglicky.

Příklady vstupů a výstupů

Pomocný soubor seznam.txt telefonního seznamu:

```
$ cat seznam.txt
Petr Dvorak
603123456
Jana Novotna
777987654
Bedrich Smetana ml.
541141120
```

Příklad hledání (pro demonstraci jsou nalezená písmena ve jméně jako velká):

```
$ ./t9search <seznam.txt
petr dvorak, 603123456
jana novotna, 777987654
bedrich smetana ml., 541141120
$ ./t9search 12 <seznam.txt
petr dvorak, 603123456
bedrich smetana ml., 541141120
$ ./t9search 686 <seznam.txt
jana nOV0tna, 777987654
$ ./t9search 38 <seznam.txt
pETr DVorak, 603123456
bedrich smETana ml., 541141120
$ ./t9search 111 <seznam.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 funkcionality vyhledání kláves a
- ošetření chybových stavů.

Priority funkcionality

1. Vyhledání kontaktů, jejichž telefonní čísla začínají zadaným číslem.
2. Vyhledání kontaktů, jejichž jména začínají znaky, které zadané číslo reprezentuje.
3. (nepovinné, avšak může zlepšit hodnocení) Hledání přerušených posloupností znaků. **Pokud jako první parametr programu bude -s (bylo přidáno 2022-10-20, není nutné toto akceptovat)**, program najde i takové kontakty, u který mezi každými dvěma nalezenými znaky může být libovolná posloupnost jiných znaků. Důležité je, že nalezený řetězec (jména nebo telefonního čísla) musí obsahovat všechny znaky, které číselky reprezentují, a ve stejném pořadí.
4. (prémiové) Hledání podobných kontaktů. Pokud zadal uživatel špatný vyhledávací řetězec (spletl se v čísle), program mu nabídne podobné výsledky. Uvažujte následující chyby uživatele: chybná číslice, chybějící číslice, číslice zadaná navíc. Program by měl být konfigurován argumentem programu -l L, kde L reprezentuje maximální počet uvažovaných chyb uživatele (pozn. symbol L je inspirován z tzv. [Levenshteinovy vzdálenosti](#) dvou řetězců).

Prémiové hodnocení

Prémiové hodnocení je dobrovolné a lze za něj získat bonusové body. Podmínkou pro udělení prémiových bodů je výborné vypracování standardního zadání. Výsledné hodnocení je plně v kompetenci vyučujícího, který bude projekt hodnotit. Výše prémiových bodů závisí také na sofistikovanosti řešení.