

Úlohy na 5. cvičenie - práca s dynamickým poľom, ukazovateľmi na funkcie, reťazce

1. Napíšte program, ktorý bude pracovať s prvkami jednorozmerného dynamického poľa. Prvkami poľa budú náhodne generované čísla z intervalu $<1, N>$.

Náhodné čísla generujte nasledovne:

použite funkciu `srand((unsigned)time(NULL))` – inicializuje generátor náhodných čísel (funkcia `time()` je definovaná v `time.h`). Funkciu `srand()` je vhodné zavolať na začiatku programu. Potom na generovanie náhodného čísla z intervalu $<0, N-1>$ použite funkciu `rand()%N`

(Např. `nahodne_cislo = rand()%(101 - 1) + 1; // generovanie nahodneho cisla z intervalu <1,100>`).

Program bude vykonávať príkazy načítané zo štandardného vstupu. Každý príkaz bude predstavovať malé písmeno nasledované koncom riadku a to:

- `g` – po aktivovaní príkazu program alokuje dynamické pole pre 100 celých čísel a naplní ho vygenerovanými číslami
- `v` – po aktivovaní program vypíše obsah poľa na štandardný výstup. Každé číslo v novom riadku
- `p` – po aktivovaní program načíta celé číslo `k` z intervalu $<1, 100>$. Následne prehľadáva pole a v prípade ak v poli nájde číslo rovnajúce sa číslu `k` na pozíciu nasledujúcu za číslom `k` pridá číslo 0. Pri tomto príkaze sa bude pole dynamicky zväčšovať. Ak v poli neexistuje načítané číslo `k` program vypíše správu `Pole sa nebude zvacsovat`. Ak pole nie je vytvorené (nebol vykonaný príkaz `g`), táto voľba nič nezmení. Pri tejto voľbe program negeneruje žiaden výstup. Výpis poľa bude prostredníctvom príkazu `v`.
- `z` – po aktivovaní program načíta celé číslo `m` z intervalu $<1, 10>$. Následne z dynamického poľa vymaže všetky prvky na pozíciách, ktoré sú deliteľné číslom `m`. Pozíciu počítajte od 0. Pri tomto príkaze sa bude pole dynamicky zmenšovať. Ak v poli neexistuje hľadaná pozícia program vypíše správu `Pole sa nebude zvacsovat`. Ak pole nie je vytvorené (nebol vykonaný príkaz `g`), táto voľba nič nezmení. Pri tejto voľbe program negeneruje žiaden výstup. Výpis poľa bude prostredníctvom príkazu `v`.
- `k` – ukončenie programu.

Všetky príkazy (okrem `k`) vykonajte v samostatnej funkcii

2. Napíšte program na zistenie počtu rôznych typov znakov v súbore `znaky.txt`. Podľa voľby zo vstupu program zisťuje počet:

- bielych znakov (medzera, tabulátor, znak konca riadku),
- veľkých písmen,
- malých písmen,
- číslíc.

Na zistenie, či je znak číslica vytvorte funkciu `int cislica(char c)`, či je malé písmeno – `int male_pismo(char c)`, veľké písmeno – `int velke_pismo(char c)`, alebo biely znak – `int biely(char c)`. Funkcie majú vrátiť hodnotu 1, ak znak je toho typu, ktorý testuje funkcia, inak vrátia 0. Ďalej vytvorte

funkciu `zisti`, ktorá bude ako argument obsahovať ukazovateľ na funkciu a ukazovateľ na súbor. Tá spočíta všetky výskyty znakov, pre ktoré vráti funkcia (ktorá bola argumentom funkcie `zisti`) hodnotu 1.

Prvý riadok vstupu bude obsahovať celé číslo n , ktoré vyjadruje počet zisťovaní. Pre každé z n zisťovaní bude na vstupe jeden riadok obsahujúci znak 'b', 'v', 'm' alebo 'c', ktorý vyjadruje, či sa zisťuje počet bielych znakov, resp. veľkých písmen, resp. malých písmen, resp. číslíc. Každý, t.j. aj posledný riadok je ukončený znakom konca riadku.

Výstupom je n riadkov, každý pre jedno zisťovanie. Pre príslušné zisťovanie bude riadok obsahovať počet zvoleného typu znakov. Každý riadok výstupu je ukončený znakom konca riadku.

Ukážka súboru `ZNAKY.TXT`:

AhOj Ferko,↵

Ako sa mas?↵

Ukážka vstupu:

2↵

v↵

b↵

Ukážka výstupu:

4↵

5↵

3. Napíšte program, ktorý bude slúžiť ako jednoduchá kalkulačka, ktorá podporuje štyri operácie: sčítanie, odčítanie, násobenie, delenie. Používateľ zadáva z klávesnice znaky vybranej operácie: `s` – sčítanie, `o` – odčítanie, `n` – násobenie, `d` – delenie a hodnoty operandov. Program sa ukončí po zadaní znaku `e` – exit. V prípade, ak používateľ zadá iný ako predtým uvedené znaky, program vypíše oznam `Operacia nie je podporovana` a čaká na zadanie ďalšej operácie. Každá operácia sa bude vykonávať v samostatnej funkcii. Jednotlivé funkcie budú volané prostredníctvom ukazovateľa na funkciu.

Ukážka vstupu:

s↵

2 5↵

t↵

n↵

3 12↵

e↵

Ukážka výstupu:

7↵

Operacia nie je podporovana↵

36↵

4. Vytvorte program na hádanie slova náhodne vybraného zo súboru `hadanka.txt`. Súbor bude obsahovať v prvom riadku počet slov. Potom bude nasledovať daný počet slov,

každé v jednom riadku vždy nasledované znakom konca riadku. Slovo reprezentujte ako pole znakov, pričom koniec slova označte znakom `'\0'`. Po načítaní slova program umožní používateľovi hádať zvolené slovo a to dvomi rôznymi spôsobmi, z ktorých si používateľ v každom kroku jeden vyberie. Tieto spôsoby sú:

- hádanie po písmenkách: používateľ zadá znak. Zobrazí sa slovo tak, že všetky doteraz uhádnuté písmená sa zobrazia (všetky výskyty týchto písmen) a namiesto neuhádnutých písmen sa zobrazí podčiarkovník.
- hádanie celého slova: má zmysel použiť, keď si už používateľ myslí, že slovo uhádol. Vtedy program načíta od používateľa celé slovo.

Program po každom kroku kontroluje, či používateľ slovo uhádol. V programe nepoužívajte funkcie pre prácu s reťazcami! Používajte indexy a ukazovatele na polia. Pre náhodné čísla použite `srand((unsigned)time(NULL))` – nastavuje náhodný generátor (funkcia `time()` je definovaná v `time.h` a funkcia `srand()` v `stdlib.h`). Funkciu `srand()` je vhodné zavolať na začiatku programu. Potom na priradenie náhodného čísla od 0 do `n-1` použite príkaz `r = (int) (N * (rand() / (RAND_MAX + 1.0)));` (funkcia `rand()` je definovaná v `stdlib.h`).

Vstup a výstup programu formátujte podľa vlastného uváženia.

Ukážka súboru `hadanka.txt`:

5↵

ahoj↵

koleso↵

slovo↵

hodiny↵

potom↵

Ukážka hádania, kde náhodne vybrané slovo je `koleso`:

(výstup) hadanie po písmenkach (p) alebo po slovach(s)?↵

(vstup) p↵

(výstup) zadajte písmeno: ↵

(vstup) o↵

(výstup) _o___o↵

(výstup) hadanie po písmenkach (p) alebo po slovach(s)?↵

(vstup) p↵

(výstup) zadajte písmeno: ↵

(vstup) e↵

(výstup) _o_e_o↵

(výstup) hadanie po písmenkach (p) alebo po slovach(s)?↵

(vstup) s↵

(výstup) zadajte slovo: ↵

(vstup) koleso↵

(výstup) Bingo!↵