

## Úlohy na 2. cvičenie – opakovanie statických polí a reťazcov

1. 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:

- *háďanie 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 podčiariovník.
- *háďanie 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.

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

(Např. `nahodne_cislo = rand()%(101 - 1) + 1;` // generovanie nahodneho cisla z intervalu  $\langle 1, 100 \rangle$ ). (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 háďania, kde náhodne vybrané slovo je `koleso`:

(výstup) háďanie po písmenkách (p) alebo po slovach(s)?↵

(vstup) p↵

(výstup) zadajte písmeno: ↵

(vstup) o↵

(výstup) \_o\_\_o↵

(výstup) háďanie po písmenkách (p) alebo po slovach(s)?↵

(vstup) p↵

(výstup) zadajte písmeno: ↵

(vstup) e↵

(výstup) \_o\_e\_o↵

(výstup) háďanie po písmenkách (p) alebo po slovach(s)?↵

(vstup) s↵

(výstup) zadajte slovo: ↵

(vstup) koleso↵

(výstup) Bingo!↵

2. Dané je pole a obsahujúce n celých čísel usporiadaných od najmenšieho po najväčšie a celé číslo `cislo`, ktoré do poľa vkladáme. Doplňte chýbajúce časti funkcie `vloz()`, ktorá vloží prvok `cislo` do poľa `a[]` tak, že výsledné pole zostane usporiadané. Pole `a[]` je dostatočné veľké aj pre ďalšie číslo.

```
void vloz(int cislo, int pole[], _____ )
{
    int i;
    for (i = _____ ; i >= 0; i--)
    {
        if ( _____ )
            break;
        pole[ _____ ] = pole[i];
    }
    pole[ _____ ] = cislo;
    (*n)++;
}
```

Správnosť doplnenia funkcie overte v programe.

Ukážka vstupu:

-45 -15 0 28 56 685↵

45↵

Ukážka výstupu:

-45 -15 0 28 45 56 685↵

3. Napíšte program obsahujúci funkcie:

(a) `void strinsert(char *s, int d, char *s2, int i)`, ktorá vloží do reťazca `s` od pozície `i` reťazec `s2`. Pre reťazec `s` je vyhradená pamäť `d` znakov. Napr. pre `s`: totojeretazec, `i`: 6 a `s2`: druhy bude po volaní funkcie `strinsert(s1, 50, s2, i)` v `s`: totojedruhyretazec. Použite funkciu `strcat` a kontrolujte, či pridaním reťazca nepresiahnete pamäť určenú pre `s`.

(b) `void strdelete(char *s, int n, char i)`, ktorá vymaže z reťazca `s` od pozície `i` `n` znakov. Kontrolujte, či je možné tieto znaky vymazať.

Funkcie použite v programe, ktorého vstupom je niekoľko riadkov. Každý riadok obsahuje riadiaci znak nasledovaný potrebnými parametrami, vždy oddelenými jednou medzerou a ukončený znakom konca riadku:

- riadiaci znak `N` nasledovaný slovom `s` znamená načítanie aktuálneho slova, s ktorým sa bude pracovať. Ak už predtým bol načítaný reťazec `s`, tento sa prepíše novým načítaným slovom. Ak ešte načítaný nebol, `s` obsahuje prázdny reťazec.
- znak `V` nasledovaný celým číslom `i` a reťazcom `s2` predstavuje príkaz na vloženie reťazca `s2` do aktuálneho reťazca `s` od pozície `i` (použitie funkcie `strinsert`).
- riadiaci znak `Z` nasledovaný dvomi celými číslami `i` a `n` znamená príkaz na vymazanie časti reťazca `s` dlhšej `n` znakov od pozície `i` (funkcia `strdelete`).
- riadiaci znak `K` znamená príkaz na ukončenie programu.  
Po každom načítanom riadku vypíše program do zvlášť riadku aktuálne slovo `s`, s ktorým pracuje. Ak nie je možné niektorý z príkazov vykonať, slovo `s`, ktoré sa nezmenilo, je nasledované jednou medzerou a jednou z chybových správ. Tieto chybové správy sú:
- do reťazca nie je možné vložiť podreťazec od zvolenej pozície (túto správu vypisujte aj v prípade, že ešte nebol načítaný aktuálny reťazec `s` a riadiaci znak určuje do neho pridávať od inej ako 0-tej pozície),

- z retazca nie je mozne vymazat znaky (túto správu vypisujete aj v prípade, že ešte nebol načítaný aktuálny reťazec s a riadiaci znak určuje z neho vymazávať).

Ukážka vstupu:

V 7 ahoj↵

N Programovanie↵

V 0 VsetciMame↵

Z 10 100↵

V 10 Radi↵

Z 0 6↵

K↵

Ukážka výstupu:

do retazca nie je mozne vlozit podretazec od zvolenej pozicie↵

Programovanie↵

VsetciMameProgramovanie↵

VsetciMameProgramovanie z retazca nie je mozne vymazat znaky↵

VsetciMameRadiProgramovanie↵

MameRadiProgramovanie↵

MameRadiProgramovanie↵