

Úlohy na 5. cvičenie – ukazovatele na funkcie, statické dvojrozmerné polia

1. Predpokladajte definíciu celočíselnej konštanty K. Napíšte funkciu

`int` porovnaj_mod(`const void` *a, `const void` *b), ktorá **porovná hodnoty**, na ktoré ukazujú ukazovatele a a b **podľa modulo K**. Ak je prvý atribút je menší podľa modulo K, funkcia vráti záporné číslo. Ak je prvý atribút väčší podľa modulo K, funkcia vráti kladné číslo. Ak sú rovnaké, vráti 0.

Porovnanie **podľa modulo K**: Ak dve čísla majú rôzny zvyšok po delení K, číslo s menším zvyškom je považované za **menšie podľa modulo K**. Ak majú rovnaké zvyšky po delení K, za **menšie podľa modulo K** sa považuje to číslo, ktoré je menšie (porovnávajú sa pôvodné čísla, nie ich zvyšky). Napr. pre čísla 10, 3, 2, 6, 12 a K = 4, sú čísla usporiadané podľa modulo K: 12, 2, 6, 10, 3, pričom ich zvyšky po delení K = 4 sú 0, 2, 2, 2, 2, 3.

Ukazovateľ na funkciu porovnaj_mod() použite v programe na usporiadanie pomocou qsort() (z knižnice stdlib.h) na usporiadanie načítaných čísel podľa modulo K.

Ukážka vstupu:

```
10 3 2 6 12↵
```

Ukážka výstupu pre K = 4:

```
12 2 6 10 3↵
```

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_pismeno(`char` c), veľké písmeno – `int` velke_pismeno(`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 v súbore, 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 vstupu je ukončený znakom konca riadku. Výstupom je n riadkov, každý pre jedno zisťovanie. Pre príslušné zisťovanie bude riadok obsahovať počet zvoleného typu znakov v súbore. 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. Vytvorte program, ktorý zistí, či je daný štvorec *magický*. Štvorec je *magický* vtedy, keď súčet prvkov vo všetkých riadkoch, stĺpcoch a uhlopriečkach je rovnaký.

Program vytvorí staticky pole $N \times N$ prvkov (napr. $N = 20$). Potom zo štandardného vstupu načíta rozmer štvorca n (pričom $n \leq N$) a do statického poľa načíta prvky štvorca. Ak bude $n < N$, nevyužije sa celý štvorec, ale len jeho časť. Ak je $n > N$, program skončí s chybovou hláškou. Potom program vo funkcii `int magicky(int pole[][N], int n)` zistí, či je štvorec magický. Nakoniec program na obrazovku vypíše správu magicky alebo nie magicky.

2 príklady magických štvorcov:

8	1	6	16	3	2	13
3	5	7	5	10	11	8
4	9	2	9	6	7	12
			4	15	14	1

4. Napíšte funkciu `void nasob(int m1[][N], int m2[][N], int m3[][N])` na vynásobenie štvorcových matíc $m1$ a $m2$ a zapíše výsledok do matice $m3$. Funkciu doplňte ju do programu z prednášky.