

# Práce s ukazateli v jazyce C

## 10 Úkol

Napiš program s podprogramy, který přečte zadaný datový soubor obsahující desetinná čísla. Na prvním řádku bude obsahovat celočíselnou hodnotu s počtem prvků, které budou následovat. Program musí být schopen zpracovat libovolně velký vstupní soubor, jehož obsah si uloží do pole, se kterým bude dále pracovat.

Program bude zapisovat výsledky do dvou uživatelsky zadaných výstupních souborů.

1. Napiš funkci, která načte libovolně dlouhý vstupní soubor do pole. Musí vrátit dynamicky alokované pole a jeho velikost. Pokud při alokaci dojde k chybě, vracej hodnotu NULL.

**Rada:** Zabal tyto dva údaje do struktury. Vypadá se to. Pak můžeš obojí vracet buď v parametru předávaném odkazem nebo dokonce jako funkční hodnotu.

**Poznámky:** Strukturu jde vracet v parametru zadávaném odkazem. Uživatelsky příjemnější je vracet ukazatel na dynamicky vytvořenou strukturu, protože to pak minimalizuje počet použití referenčních a dereferenčních operátorů v dalším kódu.

```
typedef struct {
    float *prvek; // dynamické pole
    int delka;
} Tpole;

// parametr odkazem
void nacti(FILE *vstup, Tpole *data);

// ukazatel
// alokuje strukturu i pole v ní
Tpole * nacti(FILE *vstup);
```

2. Napiš funkci, která vrátí průměrnou hodnotu načtených prvků. Výsledek vrať jako funkční hodnotu.

3. Rozděl data do dvou souborů tak, že podprůměrné hodnoty zapíšeš do prvního souboru a hodnoty průměrné a vyšší do druhého souboru. Musí být zachováno relativní vzájemné pořadí prvků. Vytvoř si pro to

funkci, která rozdělí data do dvou souborů podle zadané hranice.

```
void rozdel(FILE *f1, FILE *f2,
            Tpole *p, float hranice);
```

4. Ve funkci main dělej jen otvírání souborů a volání svých dalších funkcí (+ tisk chybových hlášení). Nakonec po sobě nezapomeň uklidit (zavřít soubory, uvolnit alokovanou paměť).

Příklad vstupního souboru:

```
11
12.5 14.3 -5.14 1.9 -7.1 15
8.5
7.94 33.77 -0.4 24.8
```

Průměrná hodnota prvků v tomto souboru je 9,642727.

Odpovídající soubor s podprůměrnými prvky:

```
6
-5.14 1.9 -7.1 8.5 7.94 -0.4
```

Odpovídající soubor s průměrnými a nadprůměrnými prvky:

```
5
12.5 14.3 15 33.77 24.8
```