

# Řazení spojového seznamu II

<b>Termín odevzdání:</b>	<b>26.12.2022 11:59:59</b>
<b>Pozdní odevzdání s penalizací:</b>	<b>31.12.2022 23:59:59</b> (Penále za pozdní odevzdání: 100.0000 %)
<b>Hodnocení:</b>	<b>5.0000</b>
<b>Max. hodnocení:</b>	<b>5.0000</b> (bez bonusů)
<b>Odevzdaná řešení:</b>	1 / 20 Volné pokusy + 10 Penalizované pokusy (-10 % penalizace za každé odevzdání)
<b>Nápovědy:</b>	0 / 2 Volné nápovědy + 2 Penalizované nápovědy (-10 % penalizace za každou nápovědu)

Úkolem je realizovat sadu funkcí (ne celý program, jen funkce) pro práci se spojovými seznamy. Funkce umožní seznam snadno vytvářet, mazat a řadit. Úloha je rozšířením jednoduššího zadání, přidává možnost vlastního komparátoru. Doporučujeme nejprve vyřešit zadání jednodušší a to pak rozšiřovat.

Při řešení problému se setkáte s následujícími deklaracemi:

## TITEM

Tato struktura reprezentuje prvek jednosměrně zřetěženého spojového seznamu. Deklarace struktury je pevně daná v testovacím prostředí, Vaše implementace bude tuto nezměněnou deklaraci používat (nemůže/nesmí ji měnit). Ve struktuře jsou následující složky:

- `m_Next` ukazatel na další prvek ve spojovém seznamu, hodnota `NULL` pro poslední prvek,
- `m_Name` řetězec s názvem prvku seznamu,
- `m_Secret` tajná data přiřazená k této položce seznamu. Vaše implementace s touto složkou nepotřebuje pracovat a nesmí ji měnit (výjimkou je inicializace ve funkci `newItem`).

## TITEM \* newItem ( const char \* name, TITEM \* next )

Tato funkce vytvoří nový záznam `TITEM`. Paměť pro strukturu budete alokovat dynamicky. Funkce navíc zkopíruje jméno a odkaz na další prvek z parametrů do složek `m_Next` a `m_Name`. Složku `m_Secret` funkce vyplní nulovými bajty (`\0`). Návratovou hodnotou funkce je ukazatel na takto vytvořenou a inicializovanou strukturu `TITEM`. Implementace funkce je Vaším úkolem.

## void freeList ( TITEM \* l )

Funkce slouží k pohodlnému výmazu spojového seznamu. Parametrem je počátek mazaného spojového seznamu `l`. Funkce zajistí uvolnění všech prostředků, které seznam zabíral. Implementace funkce je Vaším úkolem.

## TITEM \* sortListCmp ( TITEM \* l, int ascending, int (\*cmp)(const TITEM \*, const TITEM \*) )

Funkce slouží k seřazení prvků ve spojovém seznamu. Parametrem je počátek řazeného spojového seznamu `l`, požadované seřazení `ascending` a komparátor `cmp`. Funkce zajistí přeskupení prvků v zadaném seznamu tak, aby pořadí vyhovovalo požadovanému uspořádání. Funkce **nesmí** prvky původního seznamu uvolňovat (a např. vrátit jejich nově vytvořenou kopii). Naopak, **musí** přepojit odkazy existujících prvků a vrátit ukazatel na první prvek takto vzniklého seznamu.

Kritériem pro řazení je výsledek po zavolání komparátoru `cmp`. Vždy, když je potřeba určit vzájemné uspořádání dvou prvků, zavolá řadící algoritmus funkci předanou parametrem `cmp`. Volání komparátoru pracuje stejně

jako v knihovně funkci `qsort` - vrací zápornou / nulovou / kladnou hodnotu, pokud byl první parametr předaný komparátoru menší / stejný / větší než parametr druhý. Řazení dále ovlivní směr uspořádání - buď je vzestupné (parametr `ascending` není nula) nebo sestupné (parametr `ascending` je roven nule). Funkce musí zajistit, že řazení je **stabilní**.

Odevzdávejte zdrojový soubor, který obsahuje implementaci požadovaných funkcí. Do zdrojového souboru přidejte i další Vaše podpůrné funkce, které jsou z implementovaných funkcí volané. Funkce budou volané z testovacího prostředí, je proto důležité přesně dodržet zadané rozhraní funkce. Za základ pro implementaci použijte kód z příloženého archivu. Ukázka obsahuje testovací funkci `main`, uvedené hodnoty jsou použité při základním testu. Všimněte si, že vkládání hlavičkových souborů a funkce `main` jsou zabalené v bloku podmíněného překladu (`#ifdef/#endif`). Prosím, ponechte bloky podmíněného překladu i v odevzdávaném zdrojovém souboru. Podmíněný překlad Vám zjednoduší práci. Při kompilaci na Vašem počítači můžete program normálně spouštět a testovat. Při kompilaci na Progtestu funkce `main` a vkládání hlavičkových souborů "zmizí", tedy nebude kolidovat s hlavičkovými soubory a funkcí `main` testovacího prostředí.