

## Seznam zaměstnanců

<b>Termín odevzdání:</b>	<b>31.12.2017 23:59:59</b>	615646.440 sec
<b>Hodnocení:</b>	<b>0.0000</b>	
<b>Max. hodnocení:</b>	<b>5.0000</b> (bez bonusů)	
<b>Odevzdaná řešení:</b>	0 / 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 pro práci se spojovými seznamy. Spojový seznam reprezentuje seznam pracovníků ve firmě. Pracovník je reprezentován jménem a odkazem na svého zástupce. S takovým seznamem pracovníků chceme provádět tyto operace: přidání pracovníka, kopírování seznamu a mazání seznamu.

## EMPLOYEE

tato datová struktura je deklarovaná v testovacím prostředí. Vaše implementace bude s touto strukturou pracovat, ale nesmí jí nijak měnit. Struktura reprezentuje jednoho zaměstnance. Zaměstnanci jsou uloženi v podobě jednosměrně zřetěženého spojového seznamu. Struktura má následující složky:

- `m_Next` - odkaz na dalšího zaměstnance ve spojovém seznamu. Poslední zaměstnanec v seznamu má odkaz `m_Next` nastaven na hodnotu `NULL`.
- `m_Bak` - odkaz na zaměstnance - zástupce. Odkaz má buď hodnotu `NULL` (pak zástupce neexistuje), nebo odkazuje na nějaký prvek spojového seznamu. V extrémním případě může odkazovat na sebe sama, pokud v rámci úspor je zaměstnanec svým vlastním zástupem.
- `m_Name` - ASCIIIZ (nulou ukončený) řetězec udávající jméno zaměstnance.

`newEmployee ( name, next )`

funkce vytvoří nový prvek ve spojovém seznamu a umístí jej na první pozici. Parametrem je `name` - jméno nového pracovníka a `next` - odkaz na dosavadní počátek spojového seznamu zaměstnanců. Návrátovou hodnotou funkce je ukazatel na první prvek nového spojového seznamu zaměstnanců. Funkce je zodpovědná za alokaci struktury zaměstnance a za vyplnění jejích složek. Nově přidávaný zaměstnanec nemá definovaný zástup, tedy složka `m_Bak` bude mít hodnotu `NULL`.

`freeList ( list )`

funkce k uvolnění prostředků alokovaných ve spojovém seznamu zaměstnanců. Parametrem funkce je odkaz na první prvek spojového seznamu zaměstnanců dříve vytvořených pomocí funkce `newEmployee`.

`cloneList ( list )`

funkce vytvoří kopii spojového seznamu pracovníků. Nově vytvořený seznam musí zachovat jména zaměstnanců, jejich pořadí v seznamu a musí správně upravit odkazy na zástupce. Pozor, nově vytvořený seznam musí být zcela nezávislý na originále, tedy i odkazy na zástupce musí směřovat na prvky nově vytvořeného seznamu. Návrátovou hodnotou funkce je odkaz na první prvek vytvořené kopie seznamu.

Odevzdávejte zdrojový soubor, který obsahuje implementaci požadovaných funkcí. Do zdrojového souboru přidejte i další Vaše podpůrné funkce, které jsou z nich volané. Implementované funkce budou volané z testovacího prostředí, je proto důležité přesně dodržet zadané rozhraní funkcí. Za základ pro implementaci použijte kód z příloženého souboru. V kódu chybí vyplnit těla požadovaných funkcí (a případné další podpůrné funkce). 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.

Váš program bude spouštěn v omezeném testovacím prostředí. Je omezen dobou běhu (limit je vidět v logu referenčního řešení) a dále je omezena i velikost dostupné paměti. Rozumná implementace naivního algoritmu by měla projít všemi testy kromě testu rychlosti. Pro zvládnutí testu rychlosti je potřeba použít výkonnější algoritmus, který efektivně kopíruje seznamy.

## Nápověda:

- Jako základ Vašeho řešení použijte zdrojový kód z příloženého souboru.
- Do funkce `main` si můžete doplnit i další Vaše testy, případně ji můžete libovolně změnit. Důležité je zachovat podmíněný překlad.
- V ukázce je použito makro `assert`. Pokud je parametrem nenulová hodnota, makro nedělá nic. Pokud je parametrem nepravda (nula), makro ukončí program a vypíše řádku, kde k selhání došlo. Pokud tedy Vaše