

# Množina

Termín odevzdání:28.03.2021 23:59:59

Pozdní odevzdání s penalizací:16.05.2021 23:59:59 (Penále za pozdní odevzdání: 100.0000 %)

Hodnocení:0.0000

Max. hodnocení:1.0000 (bez bonusů)

Odevzdaná řešení:0 / 25

Nápovědy:0 / 25

Zbývá 1 dní, 20h, 55m a 34s

Vaším úkolem je realizovat třídu `CLinkedSet`, která bude reprezentovat abstraktní datový typ množina implementovaný seřazeným spojovým seznamem. Taková implementace je obzvláště výhodná pro operace sjednocení, průnik či rozdíl. Po vás se chce implementovat jen základní operace, splňující následující rozhraní:

konstruktor bez parametrů  
Inicializuje prázdnou množinu.

kopírující konstruktor  
Vytvoří hlubokou kopii.

operátor přiřazení  
Odstraní aktuální množinu a vytvoří hlubokou kopii předané.

destruktor  
Uvolní alokované prostředky.

```
bool Insert ( const char * value )  
Vloží do množiny řetězec, vrátí true pokud se vložení povede. Pokud již v množině řetězec je, vrátí funkce false a množinu nemění.
```

```
bool Remove ( const char * value )  
Odstraní z množiny řetězec, vrátí true pokud se smazání povede. Pokud v množině řetězec není, vrátí funkce false a množinu nemění.
```

```
bool Empty () const  
Vrátí true pokud je množina prázdná, jinak false.
```

```
size_t Size () const  
Vrátí počet prvků v množině.
```

```
bool Contains ( const char * value )  
Vrátí true pokud se řetězec nachází v množině, jinak false.
```

Řetězce jsou v seznamu seřazeny ve vzestupném pořadí, pro porovnávání využijte funkci `strcmp`.

Pro implementaci využijte strukturu `CNode`, která reprezentuje prvek ve spojovém seznamu a implementujte funkci `Value`, která vrátí řetězec.

Atribut `m_Begin` musí odkazovat na první prvek spojového seznamu a poslední prvek musí odkazovat na `nullptr`. Testovací prostředí kontroluje strukturu vašeho spojového seznamu.

**Nápověda**

- Na plný počet bodů projde řešení, které správně implementuje hlubokou kopii.
- Do třídy `CLinkedSet` a `CNode` si můžete přidat i další atributy a metody. Testovací prostředí ale spoléhá, že `m_Begin` ukazuje na začátek spojového seznamu, `m_Next` je využito jako ukazatel na následující prvek a funkce `Value` vrátí řetězec na dané pozici.
- Neimplementujte spojový seznam se zarážkou. (Pro prázdný spojový seznam musí platit, že `m_Begin = nullptr`. Pro spojový seznam s jedním prvkem `m_Begin -> m_Next = nullptr`, atd.)
- Pokud vám program nejde zkompilovat, ujistěte se, že dodržíte specifikované rozhraní.

Vzorová data:

Download

Odevzdat:

Choose File

No file chosen

Odevzdat

Referenční řešení