

Úkolem je vytvořit sadu funkcí (ne celý program, sadu funkcí), která bude implementovat databázi informací o telefonních číslech a umístění účastníků.

V úloze uvažujeme telefonní čísla v mezinárodním formátu, včetně mezinárodních kódů, například 420224351111. Číslo je tedy tvořené posloupností desítkových číslic 0 až 9 a nesmí obsahovat žádné jiné znaky. Pro takto zadané číslo lze při čtení zleva doprava snadno určit, kterému státu, regionu operátorovi číslo patří. Například prefix 4 udává, že se číslo nachází v Evropě, delší prefix 420 udává, že číslo patří do ČR. Prefix 4202 umístění zpřesňuje na Prahu v ČR. Obdobně by šlo umístění zpřesnit až ke konkrétní telefonní ústředně či účastníkovi.

Úkolem je implementovat datovou strukturu, která bude umožňovat takové vyhledávání. Vzhledem k hierarchické struktuře čísla se hodí stromová reprezentace. Strom bude 10-ární, pro každou cifru bude mít jednu úroveň, synovské uzly budou odpovídat cifrám 0 až 9. Strukturu zachycuje obrázek.



V zakresleném stromu jsou prefixy:

1	USA
1671	Guam
44	United Kingdom
420	Czech Republic
421	Slovak Republic

Vaším Úkolem je realizovat sadu funkcí, které budou umět takový strom rozšiřovat, promazávat a vyhledávat v něm. Požadované funkce mají následující rozhraní:

```
void delBook ( phonebook )
```

Funkce vymaže všechny uzly stromu a uvolní všechny alokované prostředky.

```
bool addPhone ( phonebook, phone, name )
```

Funkce přidá do telefonního seznamu předponu phone a asociuje jí s umístěním name. Funkce vrací true pro úspěch nebo false pro neúspěch (neplatné tel. číslo; v tomto případě funkce nebude nic měnit). Pokud zadaný prefix phone již v databázi existuje, funkce pouze změní asociované jméno na name a vrátí true (viz ukázkový běh s prefixem 44).

```
bool delPhone ( phonebook, phone )
```

Funkce z telefonního seznamu odstraní předponu phone. Zároveň odstraní všechny uzly stromu, které ve stromu již nemusí existovat. V extrémním případě může volání uvolnit celý strom (odstranění posledního záznamu v databázi). Návrátová hodnota je true pro úspěch nebo false pro neúspěch (neplatné tel. číslo, odstraňované tel. číslo nebylo v databázi).

```
const char * findPhone ( phonebook, phone )
```

Funkce v telefonním seznamu vyhledá umístění tel. čísla phone. Nalezené umístění vrátí jako návratovou hodnotu. Pokud pro zadané tel. číslo není známé umístění, bude vrácena hodnota nullptr. Chování funkce není zcela intuitivní, ukázkový běh demonstruje:

- výsledek nullptr pro vstupy 4 a 42 (neznámé umístění),

- výsledek Czech Republic pro vstupy 420, 4202, 42022, 420224 a 4202243 (umístění Czech Republic je nejpřesnější, které jsme schopni určit),
- výsledek Czech Republic CVUT pro vstupy 42022435, 420224351, 4202243511, ... (Czech Republic CVUT je nejpřesnější, které jsme schopni určit, umístění Czech Republic je zbytečně nepřesné).

#### TPHONEBOOK

je struktura deklarovaná v testovacím prostředí. Vaše implementace ji bude používat, ale nemůže ji měnit. Struktura obsahuje složky:

- m\_Root – ukazatel na kořen stromu s uloženými prefixy. Při inicializaci bude nastaven na hodnotu nullptr,
- m\_Size – udává počet uložených umístění, tedy počet uzlů, které mají hodnotu m\_Name nastavenou na nějaké platné jméno.

#### TNODE

je struktura deklarovaná v testovacím prostředí. Vaše implementace ji bude používat, ale nemůže ji měnit. Struktura obsahuje složky:

- m\_Name – ukazatel řetězec se jménem umístění nebo s hodnotou nullptr (umístění není zadáno),
- m\_Child – odkazy na synovské uzly, případně nullptr, pokud strom pro danou cifru dále nepokračuje.

Odevzdávejte zdrojový soubor s implementací požadovaných funkcí. Odevzdávaný soubor musí obsahovat implementaci všech požadovaných funkcí s rozhraním (parametry), které odpovídají předpisu nahoře. Dále ve zdrojovém souboru musí být další Vaše pomocné funkce, které z požadovaných funkcí voláte. V odevzdávaném zdrojovém souboru by naopak neměly být zbytečnosti (pozůstatky vývoje a ladění), vkládání hlavičkových souborů ani funkce main – toto je již obsaženo v testovacím prostředí. Pro usnadnění vývoje a odevzdávání (abyste nemuseli před každým odevzdáním ručně odstraňovat funkci main a vkládání hlaviček) použijte šablonu výše. Všimněte si, že funkce main a vkládání hlavičkových souborů je v bloku podmíněného překladu, tedy jsou testovacím prostředím přeskočeny.

Vaše funkce je testovaná v omezeném prostředí. Omezena je doba běhu i dostupná paměť. Konkrétní omezení je zřejmé z výpisu testování referenčního řešení. V této jednoduché úloze by se ale ani paměťové ani časové omezení nemělo uplatnit.

Úloha vyžaduje pečlivou práci s ukazateli a paměťovou alokací. Úloha má za úkol procvičit práci se spojovými strukturami. Testovací prostředí volá implementované funkce a zároveň kontroluje strukturu Vašeho stromu, zda odpovídá očekávání (neobsahuje zbytečné uzly, ...). Chyba je hlášena jednak pro nesprávný výsledek funkce a dále i pro nesoulad tvaru Vašeho a referenčního stromu. Podoba stromu se kontroluje po každé operaci vkládání a mazání.

Dodržte přesně rozhraní funkcí a použitých struktur. Pokud rozhraní nedodržíte, program nepůjde v testovacím prostředí zkompileovat. Zachovejte bloky podmíněného překladu jak jsou v šabloně – svévolné odstranění bloků podmíněného překladu nejspíše způsobí odmítnutí Vašeho programu.

#### Poznámky:

- V přiloženém souboru naleznete ukázkové použití funkcí, uvedené parametry jsou použité v základním testu.
- Struktura vytvořeného stromu se kontroluje po **každé** operaci přidání/odebrání. Ukázkový běh testuje strukturu stromu méně často (testy po každé modifikaci by byly příliš dlouhé).
- Platné telefonní číslo je libovolná kombinace číslic 0 až 9. Číslo může začínat číslicí 0, úvodních nul může být i více (toto je v realitě jinak, skutečná telefonní čísla v mezinárodním formátu nikdy nezačínají číslicí 0). Dvě čísla považujeme za odlišná i pokud se liší pouze v úvodních nulách (01234  $\neq$  001234).