

Druhý domácí úkol

Základy programování 1
Martin Kauer

Druhý domácí úkol 1/2

Požadavky klasicky: odevzdat jen zadané fce, žádné printy, žádný main, žádné scanf, nepoužívat žádné knihovny jako `string.h` apod.,... .

Vytvořte několik funkcí, odpovídajících následujícím deklaracím:

▶ `void invert_string (char str[]);`

V řetězci `str` nahradí malá písmena velkými a opačně.

▶ `int is_palindrome(const char str[]);`

Vrací 1, pokud je zadaný řetězec `str` palindrom, 0 pokud není.

▶ `int caesar(char str[]);`

Zašifruje řetězec `str`, použitím caesarovy šifry.

Řetězec `str` může obsahovat jen velká písmena, pokud obsahuje něco jiného, funkce musí okamžitě skončit a vrátit 0, jinak vrací 1.

Druhý domácí úkol 2/2

▶ `int remove_chars(char input[], const char remove[])`

Tato funkce smaže některé znaky v řetězci `input` a vrátí počet smazaných znaků.

Ukázky pro vstup `input = "ahhoj"`:

```
remove_chars(input, "hj"); // vrací 3, input = "ao"  
remove_chars(input, "k"); // vrací 0, input = "ahhoj"  
remove_chars(input, "a"); // vrací 1, input = "hhoj"
```

▶ `int substitute_chars(char input[], const char orig[], const char replace[])`

Tato funkce nahradí znaky v řetězci `input` za jiné a vrátí počet vykonaných nahrazení. Konkrétně nahradí každý výskyt znaku z pole `orig` znakem z pole `replace` na odpovídajícím místě. Můžete předpokládat, že pole `orig` a `replace` jsou stejně velká a písmena jsou v nich maximálně jednou.

Ukázky pro vstup `input = "ahhoj"`:

```
substitute_chars(input, "hj", "bd"); // vrací 3, input = "abbod"  
substitute_chars(input, "k", "l"); // vrací 0, input = "ahhoj"  
substitute_chars(input, "a", "_"); // vrací 1, input = "_hhoj"
```