## RLE dekomprese

Termín odevzdání: 17.12.2017 23:59:59

Hodnocení: 3.0000

Max. hodnocení: 3.0000 (bez bonusů)

Odevzdaná řešení:2 / 20 Volné pokusy + 10 Penalizované pokusy (-10 % penalizace za každé odevzdání)Nápovědy:0 / 2 Volné nápovědy + 2 Penalizované nápovědy (-10 % penalizace za každou nápovědu)

Úkolem je realizovat funkci (pouze funkci, ne celý program), která bude rozbalovat (dekomprimovat) řetězec zabalený metodou RLE.

Při realizaci budeme uvažovat velmi jednoduchou komprimační metodu RLE. Data ze zdrojového řetězce se budou buď přímo kopírovat nebo se budou kopírovat s opakováním:

- znaky zpracováváme zleva doprava,
- pokud za čteným znakem nenásleduje desítkové číslo, znak 1x zkopírujeme do výstupního řetězce,
- pokud za čteným znakem následuje desítkové číslo, přečteme toto číslo N a do výstupu zkopírujeme čtený znak N-krát. Ve
  zpracování pokračujeme dalším znakem za čteným číslem.

Vstupní řetězec tedy může být např: AB5CDE10. Pro tento vstupní řetězec bude výstup v podobě: ABBBBBCDEEEEEEEEE, tedy znaky A, C, D se pouze kopírovaly a znaky B a E se opakovaly (5x, resp. 10x).

Požadovaná funkce má rozhraní podle ukázky níže. Jejím parametrem je ASCIIZ (nulou ukončený) vstupní řetězec. Tento řetězec je určený pouze ke čtení, funkce jej nesmí nijak měnit. Návratovou hodnotou je rozbalený řetězec, který odpovídá vstupu. Prostor pro návratový řetězec musí být alokován dynamicky, volající jej uvolní voláním funkce free.

```
#ifndef __PROGTEST__
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#include <assert.h>
#endif /* __PROGTEST__ */
char * RLEDecompress ( const char * src )
  /* todo */
#ifndef ___PROGTEST___
int main ( int argc, char * argv [] )
  char * res;
  assert (! strcmp (
    (res = RLEDecompress ( "Hello world!" )),
    "Hello world!" ));
  free ( res );
  assert (! strcmp (
    (res = RLEDecompress ( "Hello 30world!" )),
                                         world!" ));
    "Hello
  free ( res );
  assert (! strcmp (
    (res = RLEDecompress ( "Hel2o world!10" )),
    `Hello world!!!!!!!!!`" ));
  free ( res );
  assert (! strcmp (
    (res = RLEDecompress ( "H2e618o15 35w5o6r-2d0!" )),
    "HHeeeeeellllllllloooooooooooo
                                                                        wwwwwooooor--!"));
  free ( res );
  assert (! strcmp (
    (res = RLEDecompress ( "321" )),
    "333333333333333333333" ));
  free ( res );
```

```
return 0;
}
#endif /* __PROGTEST__ */
```

Odevzdávejte zdrojový kód s implementací požadované funkce RLEDecompress. Použijte ukázku jako základ Vaší implementace. V ukázce je implementovaná funkce main, obsahuje řetězce použité při základním testu. Funkce je umístěna v bloku podmíněného překladu. Ten zajišťuje, že zdrojový kód můžete lokálně ladit (funkce main otestuje základní testy) a zároveň soubor můžete odevzdávat na Progtest (funkce main bude potlačena).