Johannes Kepler University Linz
LIT – Secure and Correct Systems Lab
S. Rass · S. Ahmad · M. Schwaighofer ·
K. Feichtinger · D. Zaitsev

PR Systems Programming
WS 2022
Class dates: see KUSSS

Excercise Sheet 8

Warmup (optional): Find and correct the errors (if they exist) in the following code lines:

a)

```
#include <stdio.h>
// prototype of putchar(): int putchar( int);
putchar("Z");
```

b)

```
#include <stdlib.h>
// prototype of srand(): void srand(unsigned int);
int z = srand(77U);
```

c)

```
#include <math.h>
// prototype of tan(): double tan(double);
double pi = 3.141593, angle = 31.5;
double y = tan(angle * (pi / 180.0));
```

d)

```
int printDoc(void);
printDoc("memo.txt");
```

e)

```
double average(double, double, double);
double result = average(1.7, 3.2);
```

Exercise 8.1. Define the `strInsert()` function that inserts a string at a specified position of another string. The function receives three arguments:

a) the string to insert into,

b) the string to be inserted,

c) the insertion position. The position 0 means that the second string will be inserted before the first character (as a prefix).

Hints:

- When calling the function, the `char`-array containing the first string must be large enough to store both strings! The `return` value is the new length of the first string. If the insertion position is invalid, i.e. negative or greater than the length of the first string, the return value is -1.
- A template code is attached for you to put your code into

  `strInsert-template.c`

- An example output using the template would be

  ```
  'hello ' inserted before 'Hans!':
  hello Hans!
  'my friend ' inserted at position 6 in 'hello Hans!'
  hello my friend Hans!
  ```

- You may assume that the string into which the other is inserted has enough space to take up all the extra characters. However, try out your code by allocating a small size for the string and insert another string that is much larger than the buffer. What happens?

Exercise 8.2. Write a filter program that replaces each lowercase letter in a text with the corresponding uppercase letter. The conversion is done by a bit operation. This takes advantage of the fact that a lowercase letter differs from the corresponding uppercase letter only by the 5th bit (counting from the least significant bit at index 0), is set (to 1) for uppercase letters and cleared (set to 0) for lower case letters (non-uppercase letters and other symbols should remain unchanged).

Hints:

- A filter program reads a data stream from standard input (`stdin`) and sends the manipulated data to standard output (`stdout`). Standard input and output can be redirected.
- The program shall be callable from the command line as follows:

  ```
  toupper < Orginal.txt > Result.txt
  ```

  The < in this line feeds the content of `Original.txt` into the stream `stdin`, and the > causes the stream `stdout` to write into the file `Result.txt`
- You can build your code from the skeleton file `toupper-skeleton.c`