
1 Iterieren mit Pointern

Schreiben Sie eine Funktion, die, gegeben einen String der Länge `len`, `len` Zeilen ausgibt. Die erste Zeile soll den gesamten String enthalten, die zweite Zeile soll den gesamten String *ohne den ersten Buchstaben* enthalten. Bei der dritten Zeile sollen die ersten beiden Zeichen weggelassen werden. Und so weiter bis in der `len`-ten Zeile nur noch der letzte Buchstabe des Strings steht. Der Aufruf `textTriangle("Abrakadabra")` soll also die folgende Ausgabe produzieren:

```
1 Abrakadabra
2 brakadabra
3 rakadabra
4 akadabra
5 kadabra
6 adabra
7 dabra
8 abra
9 bra
10 ra
11 a
```

Lösen Sie diese Aufgabe mittels Pointer-Arithmetik, d.h. ohne Indexierung über eckige Klammern `[]`.

```
1 void textdreieck(/*TODO*/) {
2     //TODO
3 }
4
5 int main(void) {
6     char text1[] = "Abrakadabra";
7     textdreieck(text1);
8
9     printf("\n");
10
11    char text2[] = "Hallo";
12    textdreieck(text2);
13
14    char text3[] = "";
15    textdreieck(text3);
16 }
```

2 Buchstabensuche

Ein String-Literal soll durchsucht werden, um das erste Vorkommen eines bestimmten Buchstabens `buchstabe` zu finden.

Die Funktion soll einen Pointer `string` auf ein String-Literal entgegennehmen und der Rückgabewert der Funktion ist die Position des gefundenen Buchstabens im String. Außerdem soll die Funktion über einen Rückgabeparameter `found` auch einen Pointer auf den gefundenen Buchstaben zurückgeben können. Wird der Buchstabe nicht gefunden, so soll `-1` zurückgegeben werden und der Rückgabeparameter auf `NULL` gesetzt werden.

```
1 int findeBuchstabe(char* string, char buchstabe, /*TODO*/ found) {
2     /*TODO*/
3 }
4
5 int main() {
6     char* string1 = "foo";
7     char* string2 = "bananas and ananas";
8
9     int result = findeBuchstabe(string1, 'a', NULL);
10    assert(result == -1);
11
12    result = findeBuchstabe(string2, ' ', NULL);
13    assert(result == 7);
14
15    char* charPosition;
16    result = findeBuchstabe(string2, 'd', /*TODO*/);
17    assert(result == 10);
18    assert(charPosition);
19    assert(*charPosition == 'd');
20    assert(charPosition == string2 + result);
21
22    result = findeBuchstabe(string1, 'a', /*TODO*/);
23    assert(result == -1);
24    assert(!charPosition);
25 }
```

Material

Makefile und Code-Vorlagen liegen dem beiliegenden Übungsmaterial bei. Sie können alle Binaries automatisch mittels `make` oder gezielt mit `make BINARYNAME` bauen lassen.