

## Stand

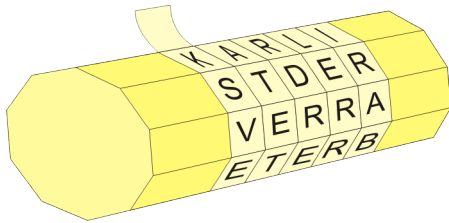
December 16, 2019

## Allgemeines

Siehe Testat 0

## Aufgabe 5: Felder / Strings

Entwickeln Sie eine Funktion zur Simulation einer Skytale (Erklärung siehe Internet).



(Bildquelle: [http://commons.wikimedia.org/wiki/File:Skytale3d\\_de.png](http://commons.wikimedia.org/wiki/File:Skytale3d_de.png))

Programmiertechnisch ist es einfacher, die "Zeilen" und "Spalten" der Skytale zu tauschen, wie <http://www.mathe.tu-freiberg.de/~hebisch/cafe/kryptographie/skytale.html> gut erläutert. Hieraus zitiert:

---

Zunächst wird der Schlüssel gewählt, was der Wahl der Dicke des Stabes der Skytale entspricht.

Beispiel  $n=3$ . zu verschlüsselnder Text DIESERKLARTEXTISTJETZTZUVERSCHLUESSELN

Zur Verschlüsselung wird der Text in Blöcke der Länge  $n$  eingeteilt, die zeilenweise geschrieben werden:

```
DIE
SER
KLA
RTE
XTI
STJ
ETZ
TZU
VER
SCH
LUE
SSE
LN
```

Damit ergibt sich spaltenweise der Geheimtext DSKRXSETVSLSLIELTTTTZECUSNERAEIJZURHEE

---

Schnittstelle:

---

```
void skytale(char * in, int n, char * out);
```

---

Bedeutung der Parameter:

**in** Der zu chiffrierende String

**n** Die Anzahl der Buchstaben auf einer Skytalenumdrehung (Schlüssel)

**out** Der chiffrierte String

Es muss nicht abgeprüft werden, ob für den Ausgabestring genügend Hauptspeicher reserviert wurde, das ist Aufgabe des Hauptprogramms. Der String wird auch nicht in Großbuchstaben transformiert oder Leerzeichen entfernt (wie im Beispiel der TU Freiberg durchgeführt)! Getestet wird aber wieder nur die Funktion, d.h. alle Funktionalität muss innerhalb der Funktion passieren. Die Main-Funktion kann daher beliebige Testfälle behandeln und beliebig ausführlich sein, sollte aber nicht zu einem Compilerfehler oder -warning führen.

- Die gesamte Kommunikation der Funktion muss über die Schnittstelle erfolgen.
- Innerhalb der Funktion sind keine Ein- und Ausgaben erlaubt.
- Es sind keine globalen Variablen erlaubt.
- Ein Prototyp der Funktion ist Pflicht.
- Gehen Sie davon aus, dass der übergebene Parameter  $n$  immer größer als 0 ist ( $n > 0$ , da ein Stift nicht weniger als die Dicke von 1 Zeichen haben darf).
- Es werden keine Leerzeichen eingefügt, auch nicht wenn die Länge des Eingabetextes nicht durch  $n$  teilbar ist!
- Verändern Sie den zu chiffrierenden Text nicht!!!!!!
- Der chiffrierte Text muss die gleiche Länge haben wie der zu chiffrierende Text!!
- Auch wenn es in dem obigen Beispiel danach aussieht, es werden keine zweidimensionalen Felder benötigt!
- Tip: Der eigentliche Chiffrier-Teil Ihres Programms ist in drei bis fünf Zeilen codierbar.
- Anmerkung: Wenn  $n$  die Schlüssellänge ist und  $l$  die Länge des Strings und  $l \bmod n = 0$ , also  $l$  ein ganzzahliges Vielfaches von  $n$  ist, so kann mit einem Dechiffrierungs-Schlüssel  $n_d = l/n$  mit demselben Algorithmus dechiffriert werden. Dies kann zum Testen des Programms genutzt werden.
- Gruppenbearbeitung: Dieses Testat darf in Gruppen von max. 3 Personen bearbeitet werden. Es muss aber JEDER Student das Testat selbst abgeben. Hierzu bitte in Ihrer Datei ganz oben als Kommentar(!) alle Gruppenmitglieder auflisten, die am Testat mitgearbeitet haben. Anschließend speichert jedes Gruppenmitglied seine Datei unter dem üblichen Namen ab und lädt sein Testat hoch.
- Abzugeben unter dem üblichen Dateinamen `<matrikel-nr>-testat-5.c` bis 13.1.2020 8:30 Uhr.  
`<matrikel-nr>-testat-5.c`