

Ausgearbeiten bis 14.01.21

**Hinweis: Verwenden Sie für die Umsetzung Funktionen****1. Primfaktorenzerlegung ( 4 Punkte)**

Schreiben Sie eine C-Funktion, die für eine gegebene Zahl eine Primfaktorenzerlegung durchführt und das Ergebnis auf dem Bildschirm ausgibt.

```
240 | 2
120 | 2
60  | 2
30   | 2
15   | 3
5    | 5
1    |
```

**2. Suche nach einer Teilzeichenkette ( 4 Punkte )**

Entwickeln Sie eine C-Programm, das alle Positionen (d.h. den Anfangsindex) ausgibt, an denen die Teilkette `pattern` in der Zeichenkette `text` vorkommt.

Beispiel:

```
text:      Dieses da ist es
```

```
pattern:    es
```

```
⇒ Positionen sind: 2, 4, 14
```

Sowohl der Text als auch die Teilkette bestehen nur aus Kleinbuchstaben und Leerzeichen.

Für die Implementierung dürfen keine Bibliotheksfunktionen verwendet werden. Berücksichtigen Sie auch die unterschiedlichen Fehlersituationen!

**3. Schrittweise Verfeinerung: *Verschlüsselung* ( 12 Punkte )**

Entwickeln Sie nach dem Prinzip der schrittweisen Verfeinerung einen Algorithmus zur Verschlüsselung eines beliebigen Textes. Der Text besteht aus Worten, die durch beliebig viele Trennzeichen (Leerzeichen, Komma und Punkt) getrennt sind.

Das Verschlüsselungsverfahren verschlüsselt den Text nach folgenden Regeln:

- Worte werden dadurch verschlüsselt, dass die Buchstaben in umgekehrter Reihenfolge (unter Beachtung von Groß- und Kleinschreibung) als „Code“ verwendet werden.

d.h. Hugo -> oguH

- Jedes 5.te Wort bleibt allerdings unverschlüsselt und wird einfach direkt in den chiffrierten Text übernommen.

- Es gibt eine Reihe von speziellen Worten, die einen festgelegten Code haben, der direkt in den chiffrierten Text übernommen wird.

Wort	Code
heute	nrets0
Bahnhof	402U
alle	hci

- Überflüssige Trennzeichen werden entfernt d.h. mehrfach vorkommende Leerzeichen, Kommas, ...
- Trennzeichen werden speziell verschlüsselt

Zeichen	Kodierungszeichen
,	@
.	#
Leerzeichen	? oder % oder &, wobei die Auswahl des Zeichens nach dem Zufallsprinzip erfolgt

Erarbeiten Sie die Lösung nach dem Prinzip der **schrittweisen Verfeinerung**! Überlegen Sie sich sinnvolle Schnittstellen und auch Datenstrukturen für eine gute Erweiterbarkeit!