

Auszuarbeiten 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!

## 1 Primfaktorenzerlegung

Lösungsidee:

Zuerst werden die Übergabeparameter überprüft. Dividor wird auf zwei gesetzt. In einer Schleife wird in jeder Iteration geprüft ob die Zahl mit dem aktuellen Dividor geteilt werden kann. Wenn ja wird die Zahl durch den aktuellen Dividor geteilt wenn nicht wird der Dividor um 1 nach oben gezählt. Das wird solange gemacht bis der Dividor kleiner gleich die Zahl ist.

Testfälle:

240; 1001; 6936; 82589933; 0

C-Program code & Test Ausgaben --> Siehe beigelegtes .c file.

## 2 Suche nach einer Teilzeichenkette

Lösungsidee:

Zuerst werden die Übergabeparameter überprüft und die Länge des Textes und des Pattern bestimmt. Dannach iterieren wir über den Eingabetext und überprüfen bei jeder Iteration ob der Anfang vom Pattern mit dem aktuellen Textbuchstabe übereinstimmt. Wenn das stimmt haben wir den Anfang von dem Pattern gefunden sonst wird normal weiter iteriert. Dannach wird mit einer weiteren Funktion überprüft ob dieses Pattern bis zum Ende stimmt. Wenn es stimmt wird die Position, vom Anfang des Patterns, ausgegeben sonst wird einfach von der Stelle, wo das Pattern falsch war, weiter gemacht.

Testfälle:

("Dieses da ist es", "es"); ("Dieses da", ""); ("Diesseess da ist eeses es", "es"); ("Dieses da ist es", " ")

C-Program code & Test Ausgaben --> Siehe beigelegtes .c file.

## 3 Schrittweise Verfeinerung: Verschlüsselung

Lösungsidee:

Zuerst wird der Übergabeparameter überprüft. Dannach wird in einer While, bei jeder Iteration, zuerst einzelne Text stücke extrahiert wobei geprüft wird ob bei der aktuellen Textposition es um ein Trennzeichen oder ein Wort handelt. Nachdem das getan ist wird das extrahierte Wort oder das extrahierte Trennzeichen codiert. Dabei wird aber, wenn es ein Wort ist, das 5te normal, ohne codierung, zum codierten Text dazu geschrieben.

Testfälle:

(" \_\_, \_\_ Der Mann, der ausgerastet ist."); (""); (" . .. , , \_ \_ ");  
("Beim Bahnhof sind heute alle mit dem Nachtexpress nach Wien mitgefahren.")

C-Program code & Test Ausgaben --> Siehe beigelegtes .c file.