technische universität dortmund

fakultät für informatik

Jurij Kuzmic
Horst Schirmeier
Jonas Ellert

Dortmund, den 11. Januar 2021

Übung zur Vorlesung EidP (WS 2020/21)
Zusatzblatt

Es können 6 Punkte erreicht werden.

Abgabedatum: 01. Februar 2021, 23:59 Uhr

Hinweise

- Bitte beachten Sie die aktuellen Hinweise unter

  `https://ls11-www.cs.tu-dortmund.de/teaching/ep2021uebung/`

- Die Aufgaben können in **Gruppen von 1 - 3 Personen** bearbeiten werden.
- Für die Abgabe sind die jeweils genannten Dateien zu erstellen.
- Stellen Sie sicher, dass alle von Ihnen abgegebene Dateien reine Textdateien im UTF-8-Format sind.
- Für die Programmieraufgaben kopieren Sie immer die Ergebnisse als Block-Kommentar an das Ende der Datei, welche das jeweilige Hauptprogramm enthält.
- Durch die Bearbeitung dieser Aufgaben können **6 Bonuspunkte** erreicht werden. Diese können einem oder mehreren der drei Blöcke der Übung auch für zurückliegende Aufgabenblätter zugeordnet werden. Dies wird von uns automatisch verteilt. Die Bearbeitung ist freiwillig.

Aufgaben

Aufgabe 1: Kryptografie durch Rotation (6 Punkte)

Die Caesar-Verschlüsselung, die auf Gaius Julius Caesar zurückgeht, verwendet für das Ver- und Entschlüsseln eine Rotation des Alphabets um n Positionen. Es gibt 25 Möglichkeiten, denn eine Rotation um 26 Positionen verändert das Alphabet nicht. Bei Beschränkung auf die 26 Großbuchstaben kann der Zielbuchstabe einer Caesar-Verschlüsselung wie folgt berechnet werden:

```
1  cout << (char)( 'A' + ( c - 'A' + (n % 26 + 26)) % 26 );
```

Dabei ist c der Buchstabe aus dem Klartext und n die Anzahl der Zeichen, um die rotiert wird. Legen Sie für Ihre Antworten die Datei `Aufgabe_1.txt` an.

a) Erklären Sie zunächst anhand eines Beispiels, warum die obige Berechung für die Großbuchstaben das richtige Ergebnis liefert. (0.6 Punkte)

b) Wie muss das n gewählt werden, damit das Ver- *und* Entschlüsseln von Großbuchstaben mit derselben Berechnung funktioniert? Wie stattdessen für Ziffern? Geben Sie Beispiele an. (0.6 Punkte)

c) Warum können mit $n = 65$ sowohl Buchstaben als auch Ziffern ver- *und* entschlüsselt werden? (0.8 Punkte)

Hinweis: Die Implementierung der Caesar-Verschlüsselung ist auf der Website der Übung zu finden. Laden Sie für Ihre Abgabe die Datei `Aufgabe_1.cpp` von der Website der Übung herunter.

d) Erweitern Sie das Programm so, dass Kleinbuchstaben und Ziffern aus dem Klartext ebenfalls verschlüsselt werden. Für die Überprüfung auf Kleinbuchstaben und Ziffern stellt die `<cctype>`-Bibliothek unter anderem Funktionen wie `islower(char)` und `isdigit(char)` zur Verfügung. (1.5 Punkte)

e) Bauen Sie das Programm um und erstellen Sie ein Menü, welches beim Start des Programms aufgerufen wird. Für diese Erweiterung sollten Sie eine Mehrfachauswahl `(switch/case)` verwenden. Der Einfachheit halber sollte dieselbe vorgegebene Berechnung `(Funktion)` sowohl für die Verschlüsselung als auch für die Entschlüsselung verwendet werden. Sie sollten beachten, dass der eingegebene Schlüssel n für das Entschlüsseln negiert werden muss. Nur dann kann dieselbe Berechnung für das Ver- *und* Entschlüsseln verwendet werden. Das Menü sollte wie folgt aussehen:

```
1  /*
2  ------------------------- Menu ------------------------------
3  Geben Sie den entsprechenden Buchstaben fuer die Funktion an.
4
5  [v] Verschluesseln
6  [e] Entschluesseln
7  [b] Programm Beenden
8
9  Auswahl [ ]:
10 */
```

Nach der Eingabe von v für das Verschlüsseln oder e für das Entschlüsseln, sollte die Eingabe von Text und Key erfolgen. Hier ist ein Beispiel für das Ver- und Entschlüsseln des Wortes „Test123!“ mit dem Schlüssel $n = 7$.

```
1  /*
2  ------------------------- Menu ------------------------------
3  Geben Sie den entsprechenden Buchstaben fuer die Funktion an.
4
5  [v] Verschluesseln
6  [e] Entschluesseln
7  [b] Programm Beenden
8
9  Auswahl [ ]: v
10 Text: Test123!
11 Key : 7
```

```
12
13 Der Text wurde erfolgreich verschluesselt!
14 Der verschluesselte Text lautet: Alza890!
15
16 ------------------------- Menu ------------------------------
17 Geben Sie den entsprechenden Buchstaben fuer die Funktion an.
18
19 [v] Verschluesseln
20 [e] Entschluesseln
21 [b] Programm Beenden
22
23 Auswahl [ ]: e
24 Text: Alza890!
25 Key : 7
26
27 Der Text wurde erfolgreich entschluesselt!
28 Der entschluesselte Text lautet: Test123!
29 */
```

(1.5 Punkte)

f) Testen Sie die Funktionsweise Ihres Programms. Ver- *und* entschlüsseln Sie die folgenden Worte und kopieren Sie die Ergebnisse als Blockkommentar an das Ende der Datei. Die Ausgabe sollte natürlich zu Ihrem Programm passen.

Verschlüsseln:

1. Text: CAESAR-VERSCHLUESSELUNG
   Key: 14
2. Text: CAESAR-VERSCHLUESSELUNG mit 39 ZEICHEN!
   Key: 65
3. Text: 15487
   Key: 7

Entschlüsseln:

1. Text: Kimaiz_Kpqnnzm
   Key: 8
2. Text: "ZXBPXO-Zefccob" jfq 65 Wbfzebk!
   Key: 23

(1 Punkt)