Informatik I: Einführung in die Programmierung

Prof. Dr. Bernhard Nebel
Tim Schulte, Thorsten Engesser
Wintersemester 2017/2018

Universität Freiburg
Institut für Informatik

Übungsblatt 7

Abgabe: Freitag, 8. Dezember 2017, 20:00 Uhr

Wichtige Hinweise: Zur Bearbeitung der Übungsaufgaben legen Sie bitte ein neues Unterverzeichnis `sheet07` im Wurzelverzeichnis Ihrer Arbeitskopie des SVN-Repositories an. Ihre Lösungen werden dann in Dateien in diesem Unterverzeichnis erwartet. Beachten Sie bitte bei allen Aufgaben die *Hinweise zur Bearbeitung der Übungsaufgaben* unter der folgenden URL:

`http://gki.informatik.uni-freiburg.de/teaching/ws1718/info1/guide/hinweise.html`

Die Abgaben zu diesem Übungsblatt werden daraufhin geprüft, ob Ihr Code PEP8-konform und Ihre Funktionen mittels Doc-Strings ausreichend dokumentiert sind. Beachten Sie hierzu die Hinweise im Python Style Guide, den Sie im Guide finden. Ab sofort wird dies für alle Abgaben vorausgesetzt.

Benutzen Sie einen Style-Checker, um alle von Ihnen eingereichten Python-Dateien daraufhin zu prüfen. Das in der Vorlesung angegebene Tool pep8 berücksichtigt nicht die in PEP8 beschriebenen Namenskonventionen für Funktionen, etc. Hierzu kann (falls nicht bereits vorhanden) das Tool flake8 und die Erweiterung pep8-naming in den meisten Python-Installationen einfach nachinstalliert werden (z.B. mit pip3 install flake8).

Aufgabe 7.1 (Microbit: Gossip; Dateien: `gossip.py`, `secrets.txt`; Punkte: 3+5)

In dieser Aufgabe dürfen Sie sich aktiv an der Verbreitung von Gerüchten beteiligen. Wir stellen Ihnen hierzu das Programm `gossip.py` auf der Kurswebsite zur Verfügung, welches Sie zunächst auf Ihren Micro Bit flashen sollen. Das Programm ermöglicht Ihnen Gerüchte, in Form geheimer Nachrichten, mit anderen Studierenden auszutauschen. Ihr Ziel besteht darin, alle in Umlauf gesetzten Gerüchte zu sammeln, um eine geheime Nachricht freizuschalten. Eine Anleitungen zum Flashen des Micro Bits finden Sie in unserem Guide[1]. Falls Sie keinen Erfolg beim Flashen haben sollten, können Sie Ihren Tutor um Hilfe bitten oder ins Betreute Programmieren gehen.

(a) Laden Sie das Programm `gossip.py` von der Kurswebsite herunter. Tragen Sie Ihren RZ-Login an der entsprechenden Stelle im Programm ein. Zum Beispiel:

```
# TODO: input your rz login name here
student = "ts112"
```

Flashen Sie das aktualisierte Programm auf Ihren Micro Bit. Commiten Sie es außerdem in das Unterverzeichnis `sheet07`.

(b) Sammeln Sie so viele Gerüchte wie möglich. Eines dieser Gerüchte erhalten Sie in Ihrer Übungsgruppe. Sie können Gerüchte von anderen Studierenden erhalten oder an sie weitergeben. Drücken Sie zum Senden Ihrer Gerüchte den A-Knopf. Gerüchte werden automatisch empfangen, allerdings müssen sich die Micro Bits hierzu sehr nah beieinander befinden. Alle so erhaltenen Gerüchte werden automatisch in einer Datei `secrets.txt` auf Ihrem Micro Bit gespeichert. Kopieren Sie diese Datei (nachdem Sie soviele Gerüchte wie möglich gesammelt haben) mit Hilfe des *Mu-Editors* vom

[1] `http://gki.informatik.uni-freiburg.de/teaching/ws1718/info1/guide/microbit.html`

Micro Bit in das Unterverzeichnis `sheet07`. Eine Anleitung zum Kopieren der Dateien finden Sie im Guide.

Aufgabe 7.2 (Kryptoanalyse; Dateien: `caesar.py`, `crack_caesar.py`; Punkte: 4+6)

Laden Sie die Datei `caesar.py` von der Webseite der Übungen zur Vorlesung herunter und speichern Sie diese im Unterverzeichnis zu diesem Übungsblatt. In dieser Datei wird die Funktion `caesar(text, shift, charset=ascii_letters)` definiert, die die Cäsarverschiebung[2] implementiert. Dabei wird in dem an die Funktion übergebenen String `text` jedes im Zeichensatz `charset` vorkommende Zeichen um `shift` viele Zeichen verschoben. Alle anderen Zeichen bleiben in der zurück gegebenen Zeichenfolge unverändert.

(a) Modifizieren Sie den Programmcode unterhalb der Zeile `if __name__ == ...` so, dass der Eingabetext von der Standardeingabe gelesen wird und das verschlüsselte Resultat in eine Datei geschrieben wird. Der Dateiname soll zusammen mit der Anzahl der zu verschiebenden Stellen (in dieser Reihenfolge) als Kommandozeilenargumente übergeben werden. Behandeln Sie Fehler (wie z.B. eine falsche Anzahl von Kommandozeilenparametern) auf für den Benutzer sinnvolle Weise. Wir vereinbaren ferner, dass die Eingabe durch eine Zeile mit dem einzigen Zeichen `.` beendet wird.

(b) Schreiben Sie ein Programm `crack_caesar.py`, das eine Cäsar-verschlüsselte Textdatei einliest und dann versucht, den enthaltenen Text automatisch und ohne Kenntnis der tatsächlichen Verschiebung zu entschlüsseln. Als einziger Kommandozeilenparameter soll der Dateiname übergeben werden. Der (im besten Fall korrekt) entschlüsselte Text soll schließlich auf der Standardausgabe ausgegeben werden. Behandeln Sie Fehler (z.B. die Angabe eines falschen oder ungültigen Dateinamens) auch hier auf für den Benutzer sinnvolle Weise.

Ihre Entschlüsselung soll hierbei die folgende Idee implementieren: die (vermutete) Entschlüsselung ist ein Text mit einer Buchstabenverteilung, die der Buchstabenverteilung in natürlichsprachlichen Texten am ähnlichsten ist (für verschiedene Sprachen finden Sie die zu erwartende Verteilung z.B. auf `https://de.wikipedia.org/wiki/Buchstabenhäufigkeit`).

Man bestimmt also zunächst für die verschiedenen (Rück-)Verschiebungen die relativen Häufigkeiten $p(c)$ der im Text vorkommenden Buchstaben $c \in \{A, \ldots, Z\}$ (Groß- und Kleinbuchstaben werden nicht unterschieden) und vergleicht diese dann mit den zu erwartenden relativen Häufigkeiten $\bar{p}(c)$ der Sprache des zu entschlüsselnden Textes (in unserem Fall gehen wir davon aus, dass der verschlüsselte Text in "Englisch" ist). Zum Vergleich zweier Häufigkeitsverteilungen verwenden Sie den sogenannten Chi-Quadrat-Test, der ein Maß für die Größe der Abweichung ist:

$$\chi^2 = \sum_{c \in \{A,\ldots,Z\}} \frac{(p(c) - \bar{p}(c))^2}{\bar{p}(c)}.$$

Das Maß ist also kleiner, je ähnlicher sich p und $\bar{p}$ sind.

Hinweis: Aus der Datei `caesar.py` können Sie in der Datei `crack_caesar.py` die Funktion `caesar` wie folgt importieren:

```
from caesar import caesar
```

[2] `https://de.wikipedia.org/wiki/Caesar-Verschlüsselung`

Aufgabe 7.3 (Erfahrungen; Datei: `erfahrungen.txt`; Punkte: 2)

Legen Sie im Unterverzeichnis `sheet07` eine Textdatei `erfahrungen.txt` an. Notieren Sie in dieser Datei kurz Ihre Erfahrungen beim Bearbeiten der Übungsaufgaben (Probleme, benötigter Zeitaufwand nach Teilaufgabe, Bezug zur Vorlesung, Interessantes, etc.).