

*Suit the action to the word,
the word to the action.*
Hamlet, William Shakespeare



UNIVERSITÄT
DES
SAARLANDES

FR Mathematik
Andreas Buchheit

10. Übung zur Vorlesung Programmierung im Sommersemester 2019

Abgabe: Mittwoch, den 26.06.2019 bis spätestens 12 Uhr.

Aufgabe 10.1. (6 Punkte) File Basics

Im Folgenden sollen das Erstellen und Bearbeiten von Dateien eingeübt werden.

- (a) (2 Punkte) Schreiben Sie eine Prozedur

```
void write_text(const char* string, const char* filename);
```

die eine neue Datei mit dem Namen `filename.txt` erstellt, und einen Strom zu ihr öffnet (bestehende Dateien gleichen Namens sollen überschrieben werden). Daraufhin soll der Inhalt des Strings `string` in die Datei geschrieben werden. Daraufhin wird der Strom wieder geschlossen.

- (b) (2 Punkte) Schreiben Sie eine Prozedur,

```
void read_text(const char* filename, size_t size, char* buffer);
```

die einen Strom zu der Datei mit dem Namen `filename.txt` öffnet und daraufhin den Inhalt in ein char Array der Länge `size` schreibt. Vergessen Sie nicht ein Null-Byte hinzuzufügen. Geben Sie eine Fehlermeldung aus, falls der Inhalt nicht in das Array passt.

- (c) (2 Punkte) Implementieren Sie Versionen der Prozeduren aus (a) und (b), die binäre Dateien lesen und schreiben.

Aufgabe 10.2. (6 Punkte) Lesen und Bearbeiten eines pgm-Bildes

In der Aufgabe zum Sierpinski-Dreieck haben wir bereits eine Bilddatei im pgm-Format erzeugt. In dieser Aufgabe soll nun ein solches Bild eingelesen, danach bearbeitet und schließlich wieder geschrieben werden.

- (a) (3 Punkte) Implementieren Sie die Funktion `read_pgm()`, die eine pgm-Datei im ASCII-Format einliest. Dieses Format ist unten beschrieben. Speichern Sie beim Lesen der Bilddatei die Anzahl der Zeilen und Spalten sowie den maximalen Grauwert ab. Die eigentlichen Bilddaten speichern Sie anschließend zeilenweise in einem langen Vektor mit Einträgen vom Typ `unsigned char`, d.h. die ersten Einträge des Vektors sind die Grauwerte der ersten Zeile des Bildes, danach kommen die Grauwerte der zweiten Zeile des Bildes u.s.w.
- (b) (3 Punkte) Erzeugen Sie nun ein kleines Hauptprogramm. Lesen Sie zunächst eine Datei ein, deren Name als erstes Argument beim Programmaufruf übergeben wird. Danach setzen Sie alle Pixel, deren Grauwert ungleich 250 ist auf den maximalen Grauwert des Bildes und Null sonst. Schließlich speichern Sie Ihr modifiziertes Bild unter dem Namen, der als zweites Argument beim Programmaufruf übergeben wurde, ab. Können Sie auf diese Weise die geheime Botschaft im Bild `secret.pgm` entziffern?

ASCII-Format von pgm-Dateien: Das pgm-Format ist wie folgt aufgebaut. In der ersten Zeile steht P2 als Kennzeichnung für das hier beschriebene Format. In der zweiten Zeile folgt die Breite und die Höhe des Bildes in Pixel. Die dritte Zeile beinhaltet den maximalen Grauwert und ab der vierten Zeile sind die Grauwerte der Pixel aufgelistet. Hierbei werden zunächst die Werte der Pixel in der ersten Zeile aufgeführt, danach die in der zweiten u.s.w. Alternativ können auch alle Grauwerte durch Leerzeilen anstatt Leerzeichen voneinander getrennt werden. Die Grauwerte befinden sich in dem Bereich von Null bis zum maximalen Grauwert aus Zeile drei und sind als Integerwerte (`unsigned char` mit Formatierung `%hhd` in `scanf`) angegeben. Von www.wikipedia.org kommt das folgende Beispiel `feep.pgm`

```
P2
24 7
15
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 3 3 3 3 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 15 15 15 0
0 3 0 0 0 0 0 7 0 0 0 0 0 0 11 0 0 0 0 0 15 0 0 15 0
0 3 3 3 0 0 0 7 7 7 0 0 0 11 11 11 0 0 0 15 15 15 15 0
0 3 0 0 0 0 0 7 0 0 0 0 0 11 0 0 0 0 0 15 0 0 0 0
0 3 0 0 0 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

