



Programmieren 3 AI – Praktikum

Übungsblatt 4

1 Punkt

Sebastian Flothow

2011-11-07

Abnahme: 2011-11-14

Zur Abnahme ist die persönliche Anwesenheit erforderlich. Bearbeiten Sie die Aufgaben so, dass Sie nicht nur funktionierenden Code vorzuweisen haben, sondern diesen auch erläutern und Fragen dazu beantworten können. Zusätzlich sind die Lösungen am Tag der Abnahme per Email an sebastian@flothow.de einzusenden; dies dient u.a. der Prüfung auf eventuelle Plagiate.

Es muss für jede Aufgabe eine getrennte Lösung vorliegen, vorzugsweise in einem separaten Verzeichnis. Falls Aufgaben die Weiterverwendung der Lösung einer vorhergehenden Aufgabe vorsehen, fertigen Sie dazu Kopien der jeweiligen Quelltexte an. Beim Kompilieren der Programme mit `g++ -Wall -Wextra -pedantic` dürfen keine Fehler oder Warnungen auftreten.

Aufgabe 1: Palindrome

Ein *Palindrom* ist ein Wort, das vor- und rückwärts gelesen gleich ist, z.B. „reliefpfeiler“ oder „reittier“. Formell können Palindrome folgendermaßen definiert werden:

- Die leere Zeichenkette ist ein Palindrom
- Jede Zeichenkette der Länge 1 ist ein Palindrom
- Eine aus mehreren Zeichen bestehende Zeichenkette, deren erstes und letztes Zeichen gleich sind, ist genau dann ein Palindrom, wenn die durch Entfernen des ersten und letzten Zeichen entstehende Teilzeichenkette ein Palindrom ist
- Jede andere Zeichenkette ist kein Palindrom

Schreiben Sie eine *rekursive* Funktion `bool isPalindrome(std::string s)`, die genau dann *true* zurückgibt, wenn `s` ein Palindrom ist. Ein rekursiver Algorithmus folgt unmittelbar aus obiger Definition. Die Funktion soll sich im Namensraum `prog3` befinden.

Schreiben Sie eine `main`-Funktion, die *wortweise* Eingaben von der Konsole liest und für jedes Wort ausgibt, ob es ein Palindrom ist. Hierfür soll die zuvor implementierte Funktion `isPalindrome` verwendet werden. Die Funktion `main` soll solange Wörter einlesen, bis das Ende der Eingabe erreicht ist.¹

Aufgabe 2: Spiegelung von Zeichenketten

Schreiben Sie ein Programm, das *zeilenweise* Eingaben von der Konsole liest. Jede gelesene Zeile soll gespiegelt, d.h. mit umgekehrter Reihenfolge der einzelnen Zeichen, wieder ausgegeben werden. Dies soll wiederholt werden, bis das Ende der Eingabe erreicht ist.

Aufgabe 3: Suchen/Ersetzen

Schreiben Sie ein Programm das zwei Befehlszeilenparameter entgegennimmt, einen Suchstring und einen Ersatzstring. Der Zugriff auf Befehlszeilenparameter erfolgt wie in C; verwenden Sie für `main` die Signatur `int main(int argc, char* argv[])`.

Das Programm soll *zeilenweise* Eingaben von der Konsole lesen, bis das Ende der Eingabe erreicht ist. In jeder gelesenen Zeile soll *jedes* Vorkommen des Suchstrings durch den Ersatzstring ersetzt werden; verwenden Sie dafür nur die Methoden der Klasse `std::string` bzw. die für `std::string` definierten Operatoren. Nach Vornahme der Ersetzung soll die Zeile wieder ausgegeben werden.

¹In der Konsole kann das Ende der Eingabe per CTRL-D signalisiert werden.