

C++ Übungsblatt 4 – Übungen zu string, vector, map

Aufgabe std::string

<http://www.cplusplus.com/reference/string/string/>

Aus der C++ Standardbibliothek bietet string eine Menge an Funktionen, die das Handhaben mit Zeichenketten vereinfachen soll.

Ziehen Sie sich für die folgenden Teilaufgaben die Dokumentation zur Hand, um möglichst elegante Lösungen zu finden. Beachten Sie, dass der Konstruktor und manche Funktionen überladen sind und so eine Auswahl an Parameter-Kombinationen anbieten, die übergeben werden können.

- a) Befehlszeilenargumente werden bei C++ Programmen in der main-Methode wie folgt erfasst

```
int main(int argc, char* argv[]) { /* ... */ }
```

argc beschreibt die Anzahl der Argumente, argv die Argumente selbst. argv[0] sollte beispielsweise den Namen der .exe-Datei angeben, die ausgeführt wird.

Diese Argumente sind allerdings vom Typ char*. Um die String-Bibliothek nutzen zu können, wandeln Sie diese Parameter in std::string um.

- b) Folgende C-Funktion prüft, ob ein Passwort richtig eingegeben wurde.

```
bool is_password_correct(char* password, char* input) {  
    return strcmp(password, input) == 0;  
}
```

Wandeln Sie diese Funktion in ein C++ Konstrukt, das anstelle von char* Parametern std::string akzeptiert.

- c) Mittels eines vom Benutzer definierten Zeichens soll ein String erstellt werden, das dieses Zeichen 50 Mal wiederholt. Die Interaktion kann ungefähr so aussehen.

```
One char please: *  
String initialized. Value is:  
*****
```

- d) Folgende Funktionsvorlage sei gegeben:

```
void add_excitement(std::string& msg, unsigned int excitement_level)
```

Der Nachricht (msg) soll hierbei je nach Höhe des excitement_level's Ausrufezeichen, Einer oder das Wort EINS bzw. ELF angehängt werden*. Bei einem excitement_level über 4 soll zusätzlich die Nachricht in Großbuchstaben umgewandelt werden.

Beachten Sie, dass der Parameter msg als Referenz übergeben wird!

* https://praxistipps.chip.de/was-heisst-1elf-das-internetphaenomen-einfach-erklaert_94397

- e) Erstellen Sie eine Funktion std::string reverse_string(std::string input), die einen übergebenen String in umgekehrter Reihenfolge zurückgibt.
- f) Der Benutzer soll einen ganzen Satz eingeben können – sehen Sie sich hierfür die Funktion std::getline() aus iostream an. Unterteilen Sie den Satz in ihre einzelnen Wörter, drehen Sie die Reihenfolge der Buchstaben um und geben Sie anschließend den Satz mit den veränderten Wörtern

aus. Satzzeichen dürfen vorerst ignoriert werden.
Beispiel: Aus „veni vidi vici“ wird „inev idiv iciv“.

Aufgabe std::vector

<http://www.cplusplus.com/reference/vector/vector/>

Dynamische Speicherallozierung war noch nie einfacher! Testen Sie den std::vector aus der Standardbibliothek aus, um bisher schwierige Aufgaben einfach zu meistern.

- a) Ein Schüler (= Benutzer) soll die Noten seiner Prüfungen eintragen. Anschließend wird seine Durchschnittsnote inklusive dem Rechenweg angezeigt. Dies kann wie folgt aussehen:

```
Prüfung 1: 1
Prüfung 2: 1
Prüfung 3: 3
Prüfung 4: 2
Prüfung 5: 4
Prüfung 6: 1
Prüfung 7:
(1 + 1 + 3 + 2 + 4 + 1) / 6 = 2
```

- b) Gegeben sei folgender Vektor:

```
std::vector<double> temperature{0.3, 0.5, 1.4, 3, 2.7, 1.2, 0, -0.1, -0.3, -0.9};
```

Die Temperaturwerte sind in Celsius angegeben. Wandeln Sie diese direkt in Fahrenheit um.
Die Umrechnungsformel hierzu lautet: $^{\circ}\text{F} = ^{\circ}\text{C} * 1.8 + 32$

- c) Aus dem temperature-Vektor der vorigen Teilaufgabe (mit den Celsius-Werten) soll ein neues gebildet werden:

- Kopieren Sie die Elemente von Index 1 (= 0.5) bis Index 8 (= -0.3) in einen neuen Vektor.
- Entfernen Sie alle negativen Zahlen im neuen Vektor.
- Fügen Sie zwischen zwei Indizes den Durchschnitt der beiden Zahlen hinzu. Beispielsweise soll bei 0.5 und 1.4 dazwischen die Zahl 0.95 eingefügt werden.

Am Ende sollte der neue Vektor folgende Zahlen beinhalten:

```
0.5, 0.95, 1.4, 2.2, 3, 2.85, 2.7, 1.95, 1.2, 0.6, 0
```

- d) Erstellen Sie ein zwei-dimensionales Vektor-Array, das char's beinhaltet. Versuchen Sie daraus ein 8x8 Schachbrettmuster zu bilden ('#' für die schwarzen, Leerzeichen für die weißen Felder) und geben Sie dieses anschließend in der Konsole aus.

```
# # # #
# # # #
# # # #
# # # #
# # # #
# # # #
# # # #
# # # #
```

Aufgabe std::map

<http://www.cplusplus.com/reference/map/map/>

Auf die Daten eines Vektors kann nur über einen Index zugegriffen werden. Will man über was anderes als eine Zahl auf Daten zugreifen, verschafft uns die Map hier Abhilfe.

- a) Erstellen Sie eine Map, die die Einwohnerzahl für bestimmte Städte beinhaltet. Fügen Sie dem ein paar Städte mit ihren Einwohnerzahlen hinzu.

Über die Kommandozeileingabe soll der Benutzer erfahren, wie viele Einwohner in der von ihm befragten Stadt leben. Ist seine angefragte Stadt nicht in der Map vorhanden, soll dem Benutzer neben der Fehlernachricht die Auswahl der Städte angezeigt werden, die abgefragt werden kann.

- b) Erweitern Sie die vorige Teilaufgabe dadurch, dass der Benutzer selber Einträge hinzufügen, löschen oder gar die gesamte Tabelle leeren kann. Zusätzlich soll der gesamte Tabelleninhalt ausgegeben werden können.