C/C++ Programmierung
Übungsblatt 2

Florian Oetke

1 Rechtsbündiges Zahlendreieck als Klasse

In dieser Aufgabe soll die Funktion `print_triangle` aus Aufgabe 3 vom letzten Übungsblatt in eine Klasse umgebaut werden, welche die Parameter kapselt und eine `void print()` Methode bereitstellt, die das Dreieck mit `std::cout` ausgibt.

Schreiben Sie hierzu eine Klasse mit den zwei privaten Member Variablen:

- `right_align_`: Boolean der wie bei Übungsblatt 1 angibt ob das Dreieck links oder rechts bündig ausgegeben werden soll. Mit Standardwert `false`.
- `size_`: Wert zwischen 0 und 9 der angibt wie breit/hoch das Dreieck werden soll, der bei der Instanziierung immer angeben werden muss.

Ihre Klasse sollte neben einer `void print()` Methode, die das Dreieck ausgibt, außerdem Getter-Methoden für alle Member Variablen bereitstellen, sowie einen Setter für `right_align_`.

Schreiben Sie außerdem eine kurze main-Funktion um ihre Klasse zu testen.

2 Zeichenketten umkehren

Schreiben Sie ein Programm das, so wie in der Vorlesung gezeigt, eine Zeichenkette aus der Standard-Eingabe einließt und sie in umkehrter Reihenfolge wieder ausgibt.

Zur Lösung dieser Aufgabe sollten sie sich vor allem nochmals mit for-Schleifen und dem Beispiel zu `std::string` auf Folie 16 von Vorlesung 1 (primär `std::ssize()` und dem Zugriff auf einzelne Zeichen) befassen.

Eingaben mit Leerzeichen müssen in dieser Aufgabe nicht berücksichtigt werden.

Bei einer Eingabe von `the-quick-brown-fox-jumps-over-the-lazy-dog` sollte die Ausgabe so aussehen:

```
god-yzal-eht-revo-spmuj-xof-nworb-kciuq-eht
```

3 Palindrome erkennen

Schreiben Sie ein Programm das, so wie in der Vorlesung gezeigt, eine Zeichenkette aus der Standard-Eingabe einließt und prüft ob es sich bei der Eingabe um ein Palindrom handelt, d.h. ob sie vorwärts und rückwärts gelesen identisch ist.

Eingaben mit Leerzeichen müssen in dieser Aufgabe nicht berücksichtigt werden.

Bei einer Eingabe von `reliefpfeiler` sollte die Ausgabe so aussehen:

```
Palindrom
```

Bei einer Eingabe von `pfeiler` sollte die Ausgabe so aussehen:

```
Kein Palindrom
```

4 Caesar-Verschlüsselung

In dieser Aufgabe soll ein sehr einfaches Verschlüsselungsverfahren implementiert werden, bei der jeder Buchstabe in der Eingabe um einen bestimmten Offset zyklisch nach rechts verschoben wird. Das heißt wenn wir z.B. `'Y'` um drei Zeichen verschieben, erhalten wir `'B'`. Andere Zeichen (Zahlen, Leerzeichen, Umlaute, Sonderzeichen, ...) sollen unverändert zurückgegeben werden.

Implementieren Sie hierzu die Funktionen `encrypt` und `decrypt` welche in der vorgegebenen main-Funktion aufgerufen werden und neben dem Eingabe-String den Offset erhalten, um den die Eingabe verschoben werden soll.

Hilfestellungen

- In C++ kann mit for-Schleifen über die Zeichen eines Strings iteriert werden (siehe Vorlesung 01)
- Einzelne Zeichen sind vom Typ `char`, mit dem wir wie mit anderen Integern rechnen können.
- Beachten Sie wann auf einer Kopie und wann ggf. mit einer Referenz gearbeitet werden muss.
- Der Wertebereich von `char` ist relativ klein. Wenn ihr Ergebnis nicht korrekt ist, ist es evtl. zu einem Overflow gekommen und sie müssen ihre Zwischenergebnisse in einem größeren Typen wie `int` speichern.

```cpp
#include <iostream>
#include <string>

// encrypt
std::string encrypt(const std::string& input, int offset) {
    // TODO
    return input;
}
// decrypt
std::string decrypt(const std::string& input, int offset) {
    // TODO
    return input;
}

void test(const std::string& message,
          const std::string& actual,
          const std::string& expected) {
    std::cout << message;

    if(actual==expected) {
        std::cout << "Korrekt: " << actual << "\n";

    } else {
        std::cout << "Falsch!\n"
                  << "  Erwartet : " << expected << "\n"
                  << "  Wert     : " << actual   << "\n";
    }
}

int main() {
    const auto input  = std::string("Test Nachricht 1");

    const auto result_4 = encrypt(input, 4);
    test("encrypt(..., 4)  : ", result_4, "Xiwx Reglvmglx 1");
    test("decrypt(..., 4)  : ", decrypt(result_4, 4), input);

    const auto result_42 = encrypt(input, -42);
    test("encrypt(..., -42): ", result_42, "Docd Xkmrbsmrd 1");
    test("decrypt(..., -42): ", decrypt(result_42, -42), input);
}
```