

Praxisorientierte Einführung in C++

Aufgabenblatt 2

Christof Elbrechter
celbrech@techfak.uni-bielefeld.de

25. April 2014

Aufgabe1: Strings umdrehen

Anstelle der parameterlosen Funktion `int main()` ist es möglich, eine Variante zu benutzen, mit der dem Programm, z.B. von der Kommandozeile aus, Argumente übergeben werden können. Diese Variante hat folgende Signatur:

```
int main(int, char *[]) /* alternativ: int main(int, char**) */
```

Der erste Parameter (vom Typ `int`) enthält die Anzahl der Kommandozeilenargumente. Der zweite Parameter (vom Typ `char *[]`), stellt ein Array von Zeigern auf C-Strings dar. Damit kann man z.B. auf folgende Weise auf das erste Kommandozeilenargument zugreifen (und es, in diesem Falle, auf der Konsole ausgeben):

```
#include <iostream>

int main(int argc, char *argv[]) {
    std::cout << "Programmname: " << argv[0] << std::endl;
    if(argc > 1){
        std::cout << "Erstes Argument: " << argv[1] << std::endl;
    }
}
```

- (2 Punkte) Schreiben Sie ein Programm, welches alle Kommandozeilenparameter in umgekehrter Reihenfolge auf der Konsole ausgibt.
- (2 Bonus Punkte) Nun sollen die Argumente in richtiger Reihenfolge ausgegeben werden. Benutzen Sie anstelle des Indexoperators `[]` Zeigerarithmetik, um die Kommandozeilenparameter abzuarbeiten. Benutzen Sie hierbei auch die Tatsache, daß als letzter Eintrag in `argv` (also an der Speicheradresse `argv[argc]`), ein 0-Pointer steht. Verwenden Sie `argc` gar nicht.
- (3 Punkte) Schreiben Sie ein Program, welches einen String als Programmargument erwartet und ihn in umgekehrter Reihenfolge auf der Konsole ausgibt. Beispiel: "Anna" soll als "anna" ausgegeben werden. Denken Sie daran, dass das erste Argument von der Kommandozeile als *zweites* Programmargument in `main` zur Verfügung steht.

Aufgabe2: Sortieren

- a. (3 Punkte) Schreiben Sie ein Programm, welches ein Array sortiert. Fragen Sie hierfür vom Benutzer unter Verwendung von `std::cin` 7 `int`-Zahlen ab. Legen Sie diese in einem Array vom Typ `float` [7] ab und geben Sie sie in aufsteigender Reihenfolge auf der Konsole aus. Welchen Sortieralgorithmus verwenden Sie?

Hinweise:

- Um, zu testen, ob eine Zahl größer als eine andere ist, können Sie z.B. eine `if`-Abfrage folgender Form verwenden:

```
if (x < y) {  
    // Die hier stehenden Anweisungen werden nur ausgeführt,  
    // wenn x echt kleiner als y ist  
}
```

- Wenn Sie wollen, können Sie sich überlegen, ob und in welcher Form man die in der Vorlesung angesprochene Pointerarithmetik hierfür verwenden kann.
- Die Definition von `std::cin` befindet sich im Header `iostream`. Hier ein Beispiel für die Verwendung von `std::cin`:

```
#include <iostream>  
int main(){  
    int a;  
    std::cout << "bitte ein int eingeben:" << std::flush;  
    std::cin >> a;  
    std::cout << "es wurde " << a  
                << " eingegeben" << std::endl << std::endl;  
    char cs[3]={0};  
    std::cout << "bitte 2 chars eingeben:" << std::flush;  
    std::cin >> cs[0] >> cs[1];  
    std::cout << "es wurden " << cs  
                << " eingegeben" << std::endl << std::endl;  
}
```

- b. (2 Punkte) Implementieren Sie das Programm erneut, verwenden Sie aber dieses mal die Funktion `std::sort`, welche in dem Header `<algorithm>` definiert ist.
- c. (3 **Bonus**punkte) Implementieren Sie das Programm nun ein drittes mal. Nun soll allerdings die C-Funktion `qsort` verwendet werden. Suchen Sie im Internet eine Beschreibung von `qsort`. Was ist der Unterschied zu `std::sort`? Welche Funktion sollte schneller sein?

Aufgabe3: Parser

- a. (4 Punkte) Schreiben Sie ein Programm, welches die Sprache $c^*a^ib^ja^ib^i$ akzeptiert. Das zu parsende Wort soll als erstes Programmargument übergeben werden. Die Ausgabe des Programms sollte "OK" lauten, wenn das Wort akzeptiert wurde und "Nicht OK" falls nicht. Optional könnte das Programm bei einem Fehler eine zweite Zeile ausgeben, die angibt, welcher Buchstabe nicht mehr ins Schema passte.