

Übung zur Vorlesung EidP (WS 2019/20)

Blatt 5

Block rot

Es können 4 Punkte erreicht werden.

Abgabedatum: 21. November 2019, 23:59 Uhr

Hinweise

- Bitte beachten Sie die aktuellen Hinweise unter

<https://ls11-www.cs.tu-dortmund.de/teaching/ep1920uebung/>

- Für die Abgabe sind die jeweils genannten Dateien zu erstellen.
- Stellen Sie sicher, dass alle von Ihnen abgegebene Dateien reine Textdateien im UTF-8-Format sind.
- Für die Kompilierung des Programms muss der C++14-Standard aktiviert sein. Dies kann im Referenzcompiler GCC 6.3 durch den Schalter `-std=c++14` sichergestellt werden. Es sollen zudem die Parameter `-pedantic` und `-Werror` genutzt werden. Der Befehl zum Kompilieren soll somit wie folgt aussehen:

```
g++-6 -pedantic -Werror -std=c++14 Aufgabe.cpp -o Aufgabe
```
- Die Verwendung von zusätzlichen **Bibliotheken** zur Lösung der Aufgaben ist **nicht erlaubt!**

Aufgaben

Aufgabe 1: Grundlagen (0.5 Punkte)

Legen Sie für Ihre Antworten eine Textdatei `Aufgabe_05_1.txt` an.

- a) Was versteht man unter einer Funktionsschablone? Erläutern Sie die Definition einer Funktionsschablone anhand eines Beispiels. (0.1 Punkte)
- b) Was ist bei der Definition einer Funktion zu beachten, der ein mehrdimensionales Array übergeben werden soll? Beachten Sie dabei statische und dynamische Arrays. Geben Sie Beispiele an. (0.2 Punkte)

c) Wie entscheidet der Compiler, welche Funktion er auf Basis einer Funktionsschablone erzeugen soll? (0.1 Punkte)

d) Was sind inline-Funktionen und wo liegt deren Vor- bzw. Nachteil zu „normalen“ Funktionen? (0.1 Punkte)

Aufgabe 2: Funktionen und Arrays (1 Punkt)

Ergänzen Sie das gegebene Codefragment. Legen Sie dazu die Datei `Aufgabe_05_2.cpp` an.

a) Schreiben Sie eine Funktion `amul`, welche ein Array von `int`-Werten und deren Länge übergeben bekommt. Zudem soll ein Integerwert `s` übergeben werden. Die Einträge des Arrays sollen mit `s` multipliziert werden. Als Rückgabe soll die Funktion das übergebene Array zurückgeben. (0.5 Punkte)

b) Schreiben Sie eine Funktion mit Namen `aadd`, welche zwei Arrays von `int`-Werten und deren Längen übergeben bekommt. Die Funktion soll die beiden Arrays stellenweise addieren und das Ergebnis im ersten Array speichern. Falls jedoch die Längen der beiden Arrays unterschiedlich sind, soll keine Addition durchgeführt werden. Als Rückgabe soll die Funktion das erste übergebene Array zurückgeben. (0.5 Punkte)

```
1  /** Aufgabe_05_2.cpp **/
2  #include <iostream>
3  using namespace std;
4
5  //*****
6  // Ergaenzen: Definition der Funktionen
7  // amul und aadd
8  //*****
9
10 int main() {
11     unsigned int const n = 5;
12     int a1[n] = { 5, 2, 4, 7, 2 };
13     int a2[n] = { 6, 3, 3, -1, 9};
14
15     amul(aadd(aadd(a1, n, a2, n), n, a2, n), n, 2);
16
17     cout << "Ergebnis: [ ";
18     for (unsigned int i = 0; i < n - 1; ++i) {
19         cout << a1[i] << ", ";
20     }
21     cout << a1[n - 1] << " ]" << endl;
22     return 0;
23 }
24 /* Ausgabe:
25 */
26 /** Ende Aufgabe_05_2.cpp **/
```

Aufgabe 3: Funktionsschablonen (1 Punkt)

Ergänzen Sie das gegebene Codefragment. Legen Sie dazu die Datei `Aufgabe_05_3.cpp` an.

a) Nutzen Sie Schablonen, um eine Funktion `rotate` zu schreiben, welche ein Array eines beliebigen Typs übergeben bekommt. Jeder Eintrag des Arrays soll um eine Position nach links verschoben werden, wobei der Eintrag der ersten Position an die letzte Position verschoben wird. (0.5 Punkte)

b) Schreiben Sie eine Funktion mit Namen `pprint`, welche zwei Werte möglicherweise unterschiedlichen Typs übergeben bekommt. Die Funktion soll die beiden beiden Werte in folgendem Format ausgeben `[Wert1 : Wert2]`. (0.5 Punkte)

```
1  /** Aufgabe_05_3.cpp */
2  #include <iostream>
3  using namespace std;
4
5  //*****
6  // Ergaenzen: Definition der Funktionen
7  // rotate und pprint
8  //*****
9
10 int main() {
11
12     unsigned int const n = 5;
13     int a1[] = { 0, 1, 2, 3, 4 };
14     char a2[] = { 'a', 'V', 'R', 'd', 'r' };
15
16     rotate(a1, n);
17     cout << "Ergebnis: [ ";
18     for (unsigned int i = 0; i < n - 1; ++i) {
19         cout << a1[i] << ", ";
20     }
21     cout << a1[n - 1] << " ]" << endl;
22
23     rotate(a2, n);
24
25     cout << "Ergebnis: [ ";
26     for (unsigned int i = 0; i < n - 1; ++i) {
27         cout << a2[i] << ", ";
28     }
29     cout << a2[n - 1] << " ]" << endl;
30
31     pprint(1.0, "Hello");
32     pprint('x', 42);
33
34     return 0;
35 }
36 /* Ausgabe:
37 */
38 /** Ende Aufgabe_05_3.cpp */
```

Aufgabe 4: DEA (1.5 Punkte)

Legen Sie für Ihre Antworten die Textdatei `Aufgabe_05_4.txt` an.

Gegeben sei der **DEA** $A = (S, \Sigma, \delta, F, s_0)$ mit $S = \{1, 2, 3, 4, 5\}$, $\Sigma = \{x, y\}$, $F = \{2, 4\}$, $s_0 = 1$ und δ gemäß folgender Tabelle:

δ	x	y
1	2	1
2	3	5
3	4	2
4	2	4
5	1	1

Welche der folgenden beiden Eingaben werden von dem Automaten akzeptiert? Geben Sie für beide Eingaben auch den Zustand an, in dem sich der Automat zu Beginn und nach Lesen jedes Zeichens befindet.

i) $xyyy$

ii) $yyxxxx$

Geben Sie außerdem die kürzeste Eingabe an, die in einem Finalzustand endet und alle Zustände vorher besucht hat. Begründen Sie Ihre Antwort.

Präsenzaufgabe 5: DEA (0 Punkte)

Entwerfen Sie einen **DEA** mit dem Eingabealphabet $\Sigma = \{a, b, 1\}$, der genau dann akzeptiert, wenn die Eingabe die Zeichenkette \mathbf{ball} enthält. Geben Sie dabei die Tabelle für δ , die Zustandsmenge S , den Startzustand s_0 und die Menge der Finalzustände F an. Dabei soll z. B. die Eingabe $\mathbf{baballa}$ akzeptiert werden, die Eingabe \mathbf{baall} jedoch nicht.