

Praktikum zur Vorlesung Einführung in die Programmierung WS 20/21

Blatt 4

Es können 15 (+ 6) Punkte erreicht werden.

Allgemeine Hinweise

1. Bitte lesen Sie vor der Bearbeitung **alle** Aufgaben sorgfältig durch! Dies erspart Ihnen unnötige Arbeit und somit auch Zeit!
2. Die einzigen Header, die Sie zur Bearbeitung der Aufgaben verwenden dürfen, sind `iostream` und solche, die laut Aufgabenstellung explizit erlaubt werden.
3. Lassen Sie sich fertiggestellte Aufgaben bitte möglichst **frühzeitig** testen.
4. Wir akzeptieren ein Testat nur, wenn die Lösung eigenständig auf Anhieb erklärt werden kann. Andernfalls müssen wir die entsprechende Teilaufgabe mit 0 Punkten bewerten.

Aufgabe 1: Benutzen des Debuggers (8 Punkte)

Laden Sie sich von der Praktikums-Webseite die Anleitung für die Verwendung des Debuggers herunter und lesen Sie diese. Erstellen Sie anschließend ein neues Verzeichnis mit dem Namen `Aufgabe_4_1` und fügen Sie die zur Verfügung gestellte C++-Quelldatei `debug.cpp` hinzu.

a) Starten Sie das Programm im Debugging-Modus. Setzen Sie in der Programmzeile bei der Definition der Variablen `andereZahl` der Funktion `main()` einen Haltepunkt. Gehen Sie dann das Programm schrittweise durch. Angenommen, Zeile x wird gerade durch einen Pfeil und besondere Hintergrundfärbung hervorgehoben. Welche Zeilen des Programms wurden in diesem Moment schon ausgeführt? Wie verhält sich der Debugger bei Bedingungen? Welche Zeilen werden ausgewertet?

_____ (3)

b) Setzen Sie einen Haltepunkt in der Zeile mit der Anweisung `cout << "Zahlenvergleich" << endl;`. Welchen Wert haben die Variablen `eineZahl`, `andereZahl`, `text1` und `text2`, wenn das Programm an diesem Haltepunkt anhält?

_____ (1)

c) Behalten Sie den eben genutzten Haltepunkt bei. Klicken Sie mit der rechten Maustaste in dem Reiter **VARIABLES** auf die Variable `eineZahl` und wählen Sie die Option **Set Value**. Geben Sie nun einen Wert für die Variable ein, drücken Sie Enter und lassen das Programm weiter laufen. Führen Sie das Programm mehrfach aus und probieren Sie hier unterschiedliche Werte aus. Wie verhält sich das Programm für die unterschiedlichen Werte?

_____ (2)

d) Legen Sie in dem Programm zwei Variablen vom Datentyp `char` an und weisen Sie ihnen die Werte `'q'` und `'&'` zu. Welche Informationen lassen sich mit dem Debugger über die Repräsentationen von `'q'` und `'&'` gewinnen?

_____ (2)

Aufgabe 2: Wurzelziehen (7 Punkte)

Erstellen Sie zunächst ein neues Verzeichnis `Aufgabe_4_2` und darin eine C++-Quelltextdatei `wurzel.cpp`.

a) Schreiben Sie eine Funktion `wurzel`, die eine Zahl a vom Typ `double` als Eingabe erhält und die berechnete Wurzel als `double` zurückgibt.

Diese Funktion soll das Heron-Verfahren implementieren, welches die Quadratwurzel \sqrt{a} iterativ annähert. Das Heron-Verfahren ist definiert durch einen Startwert

$$x_0 = \frac{a + 1}{2}$$

und einen Näherungsschritt

$$x_{n+1} = \frac{1}{2} \left(x_n + \frac{a}{x_n} \right).$$

Die Variable $x_i \approx \sqrt{a}$ ist dabei der Näherungswert für die Quadratwurzel in Iteration i . Setzen Sie innerhalb der Funktion eine Schleife ein, die abbricht, sobald $x_n = x_{n+1}$, spätestens aber nach 100 Iterationen. Bei jeder Iteration soll die aktuelle Iteration i sowie der aktuelle Wert von x_n in der Form `x = 3, i = 42` ausgegeben werden. Die Anweisung `cout << setprecision(n);` (mit integer-Argument n) legt die Anzahl der auf dem Bildschirm ausgegebenen Stellen (insgesamt vor und nach dem Komma) fest. Hiermit können Sie die Anzahl der ausgegebenen Nachkommastellen bestimmen.

_____ (3)

b) Schreiben Sie eine `main`-Funktion, welche den errechneten Wert von \sqrt{a} , das Ergebnis der Funktion `sqrt`, sowie die Differenz `sqrt(a) - wurzel(a)`, mit 17 Stellen auf dem Bildschirm ausgibt. Versehen Sie die Ausgabe bitte jeweils mit einem kleinen Text, wie z.B. `die Differenz beträgt:`. Ihr Programm soll die Wurzel von 256512.698 berechnen.

Zur Berechnung der Wurzel mit \sqrt{a} binden Sie zusätzlich zur Bibliothek `iostream` die Bibliothek `iomanip` sowie `cmath` ein.

_____ (2)

c) Die Funktion `wurzel` soll nun einen weiteren Parameter vom Typ `int` erhalten. Dieser soll als Obergrenze für die Anzahl an Schleifendurchläufen genutzt werden.

_____ (1)

d) Starten Sie anschließend Ihr Programm und notieren Sie sich die ausgegebenen Werte gemäß der untenstehenden Tabelle.

Eingabe	Anzahl Schleifendurchläufe	Ergebnis <code>wurzel(a)</code>	Ergebnis <code>sqrt(a)</code>	Differenz
2				
8				
12				
20				

_____ (1)

Aufgabe 3: Anagramme (Optional) (6 Punkte)

Erstellen Sie zunächst ein neues Verzeichnis `Aufgabe_4_3` und fügen Sie die zur Verfügung gestellte C++-Quelldatei `anagramme.cpp` hinzu.

Ein Wort ist ein *Anagramm* eines anderen Wortes, wenn beide Wörter dieselben Buchstaben mit jeweils derselben Anzahl enthalten. Beispielsweise ist „Lager“ ein Anagramm von „Regal“, weil beide jeweils aus einem A, einem E, einem G, einem L und einem R bestehen. Im Gegensatz dazu sind „Banane“ und „Bannen“ keine Anagramme; sie unterscheiden sich in der Anzahl der A und der N.

In dieser Aufgabe soll ein Programm geschrieben werden, das feststellt, ob zwei vorgegebene Wörter Anagramme voneinander sind. Dazu wird in einem `int`-Array der Länge 26 zunächst gezählt, wie häufig jeder Buchstabe in einem der Wörter vorkommt. Anschließend wird in demselben Array für jeden im zweiten Wort vorkommenden Buchstaben wieder heruntergezählt. Es handelt sich dann um Anagramme, wenn am Ende jeder Eintrag im Array 0 ist.

Im Rahmen dieser Aufgabe soll Groß-/Kleinschreibung ignoriert werden (wie im Beispiel Lager/Regal). Außerdem dürfen Sie davon ausgehen, dass die übergebenen Wörter nur Buchstaben enthalten (von a bis z und von A bis Z, also keine Ziffern, Sonderzeichen, Umlaute oder sonstige Buchstaben). Eine gesonderte Fehlerbehandlung für Wörter mit anderen Zeichen ist nicht gefordert.

a) Schreiben Sie eine Funktion `position` mit Rückgabety `int`, die für einen übergebenen Buchstaben (Typ `char`) zurückgibt, an welcher Position des Alphabets er steht. Für den Buchstaben A (also 'a' oder 'A') soll 0 zurückgegeben werden, für B ('b' oder 'B') soll 1 zurückgegeben werden, und so weiter.

HINWEIS: Eine einzige `if`-Anweisung ist hierfür ausreichend.

_____ (1)

b) Schreiben Sie eine Funktion `zaehleHoch` mit einem Parameter `buchstabe` vom Typ `char` und einem Parameter `anzahlen` vom Typ `int []`. Innerhalb der Funktion soll der Eintrag des `anzahlen`-Arrays, der der Position des übergebenen Buchstaben entspricht, um eins erhöht werden. Für `buchstabe == 'a'` soll also beispielsweise das Element mit Index 0 inkrementiert werden. Die Funktion soll nichts zurückgeben.

Schreiben Sie außerdem eine Funktion `zaehleRunter` mit derselben Signatur und demselben Rückgabety `int []` wie `zaehleHoch`. Die Funktion soll analog zu `zaehleHoch` den entsprechenden Eintrag im Array um eins verringern.

_____ (2)

c) Schreiben Sie eine Funktion `zaehleAlleHoch`, die die obige Funktion `zaehleHoch` für jeden Buchstaben eines Wortes aufrufen soll. Dabei soll immer dasselbe `anzahlen`-Array verwendet werden, das der Funktion `zaehleAlleHoch` geeignet übergeben wird. Auch das Wort soll der Funktion geeignet übergeben werden.

Schreiben Sie außerdem eine Funktion `zaehleAlleRunter`, die analog für jeden Buchstaben eines Wortes die Funktion `zaehleRunter` aufruft.

_____ (2)

d) Schreiben Sie eine Funktion `sindAlleNull` mit Rückgabety `bool`, einem Parameter `anzahlen` vom Typ `int []` und einem Parameter `laenge` vom Typ `int`. Der Parameter `laenge` soll dabei für die Länge des übergebenen Arrays stehen. Die Funktion soll `true` zurückgeben, wenn alle Einträge des Arrays gleich 0 sind, und `false` sonst.

_____ (1)