technische universität dortmund

fakultät für informatik

Amer Krivošija
Denis Kurz
Eugen Rusakov

Dortmund, den 3. November 2016

Übung zur Vorlesung EidP (WS 2016/17)

Blatt 2

Block gelb

Es können 4 Punkte erreicht werden.

Abgabedatum: 10. November 2016, 23:59 Uhr

Hinweise

- Bitte beachten Sie die aktuellen Hinweise unter

  `https://ls11-www.cs.tu-dortmund.de/teaching/ep1617uebung/`

- Für das vorliegende Übungsblatt sind letztmalig Einzelabgaben erlaubt. Ab dem Übungsblatt 3 werden nur noch Gruppenabgaben akzeptiert.
- In der optionalen Datei `Anmerkungen.txt` können Sie allgemeine Anmerkungen bezüglich Ihrer Lösungen notieren.
- Die Aufgaben müssen bis zum oben angegebenen Abgabetermin abgegeben werden über das Web-Interface unter

  `https://ess.cs.tu-dortmund.de/ASSESS/`

  Dieses Programm fragt nach Informationen über die Gruppenteilnehmer (**Vor-, Nachnamen, Matrikelnummern**) und sammelt im aktuellen Verzeichnis die abzugebenden Dateien ein. Namen und Anzahl von abzugebenden C++-Quellcodedateien variieren und stehen in der jeweiligen Aufgabenstellung. Wählen sie dazu **Abgabe** und als Veranstaltung **EidP Übungen WS 2016/17** aus und laden Sie Ihre Ergebnisdateien hoch. Bis zum Abgabetermin kann eine Aufgabe beliebig oft abgegeben werden. Bewertet wird immer die letzte von Ihnen hochgeladene Version.
- Sobald eine Abgabe von den Betreuern korrigiert wurde, können die erzielte Punktzahl und die korrigierte Lösung ebenfalls unter dieser Adresse eingesehen werden.
- Es ist ratsam, die Programme vor der Abgabe zu kompilieren und auszuführen.
- Verwenden Sie für die Textaufgaben reine Textdateien. Das Abgabesystem erkennt **keine** Word- oder PDF-Dateien!
- Die Aufgaben sind in Zweier- oder Dreiergruppen zu bearbeiten. Der Lösungsweg und die Programmierung sind gemeinsam zu erarbeiten.

- Die Gruppenmitglieder sollten gemeinsam an der gleichen Übungsgruppe teilnehmen. Die Lösung wird jeweils komplett bewertet und den Gruppenmitgliedern gleichermaßen angerechnet.
- Die Übungsaufgaben müssen spätestens bis zum jeweiligen Abgabetermin (siehe jeweiliger Aufgabenzettel) abgegeben werden. In darauffolgenden Übungen sind teilweise einzelne abgegebene Lösungen oder auch eine Musterlösung zu präsentieren und zu besprechen.
- Die abgegebenen Antworten/Programme werden automatisch auf Ähnlichkeit mit anderen Abgaben überprüft. Werden hierbei starke Übereinstimmungen festgestellt, wird die Aufgabe als nicht abgegeben bewertet.

Aufgaben

Aufgabe 0: Grundlagen (1 Punkt)

Speichern Sie Ihre Antworten zu den folgenden Fragen in der Textdatei `Aufgabe_02_0.txt`.

(a) Welche zusammengesetzten Datentypen haben Sie bisher kennengelernt? (0,2 Punkte)

(b) Welche Eigenschaften hat eine Aufzählung? (0,2 Punkte)

(c) Es sei das zweidimensionale Array `A` vom Typ `int` mit der Größe 11 mal 5 gegeben. Welche Indizes gehören zum Element von `A`, das in der letzten Zeile und der letzten Spalte ist? (0,2 Punkte)

(d) Was wird durch `'\0'` gekennzeichnet? (0,2 Punkte)

(e) Wie ist die Datendefinition eines Datenverbunds (`struct`) festgelegt? (0,2 Punkte)

Aufgabe 1: Statistiken (3 Punkte)

Sei $A = (a_1, \ldots, a_n)$ eine Folge von reellen Zahlen. Mit $\bar{a}_{sum}$ definieren wir die *gewichtete Summe* dieser Zahlen, wobei a_i mit $\frac{1}{i}$ gewichtet werden soll:

$$\bar{a}_{sum} = \sum_{i=1}^{n} \frac{a_i}{i} = \frac{a_1}{1} + \frac{a_2}{2} + \ldots + \frac{a_n}{n}$$

Das *harmonische Mittel* $\bar{a}_{harm}$ dieser Zahlen definieren wir als:

$$\bar{a}_{harm} = \frac{n}{\sum_{i=1}^{n} \frac{1}{a_i}} = \frac{n}{\frac{1}{a_1} + \frac{1}{a_2} + \ldots + \frac{1}{a_n}}$$

Der Rang eines Elements von A ist seine Position in der *sortierten* Reihenfolge von A. Das kleinste Element von A hat also Rang 1, das zweitkleinste hat Rang 2 und das größte hat Rang n. Wir bezeichnen mit $a_{(i)}$ das Element aus A mit Rang i. Die Folge $(a_{(1)}, \ldots, a_{(n)})$ enthält also dieselben Elemente wie A, aber in sortierter Reihenfolge. Der Median von A ist das Element $a_{(\frac{n+1}{2})}$, falls n ungerade ist. Für gerade n ist der Median von A das arithmetische Mittel von der Elemente mit Rang $\frac{n}{2}$ und $\frac{n}{2} + 1$, also $\frac{1}{2}(a_{(\frac{n}{2})} + a_{(\frac{n}{2}+1)})$.

Beispiel: Für $A = (5, 9, 3, 2, 7)$ (und damit $n = 5$) gilt $a_{(1)} = 2$, $a_{(2)} = 3$, $a_{(3)} = 5$, $a_{(4)} = 7$ und $a_{(5)} = 9$. Der Median dieser Folge ist 5. Die gewichtete Summe $\bar{a}_{sum}$ beträgt 12,4, das harmonische Mittel $\bar{a}_{harm}$ ungefähr 3,884.

Beispiel: Für $A = (5, 9, 3, 7)$ (und damit $n = 4$) gilt $a_{(1)} = 3$, $a_{(2)} = 5$, $a_{(3)} = 7$ und $a_{(4)} = 9$. Der Median dieser Folge ist das arithmetische Mittel aus 5 und 7, also 6. Die gewichtete Summe $\bar{a}_{sum}$ beträgt 12,25, das harmonische Mittel $\bar{a}_{harm}$ ungefähr 5,081.

(a) Legen Sie eine Datei `Aufgabe_02_1a.cpp` an. Ergänzen Sie darin das folgende C++-Programmfragment so, dass bei Ausführung des Programms der Variable `sum` der Wert der gewichteten Summe und der Variable `harm` der Wert des harmonischen Mittels der durch das Array `A` dargestellten Zahlenfolge zugewiesen wird. (1 Punkt)

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      double A[] = { 1.03, 94.62, 94.87, 88.97, 51.64,
6                     13.28, 85.8, 75.07, 23.23, 46.08 };
7      int n = sizeof(A) / sizeof(double);
8      double sum;
9      double harm;
10
11     //************************************
12     // Hier beliebig viele Zeilen ergaenzen
13     //************************************
14
15     cout << "Gewichtete Summe: " << sum << endl;
16     cout << "Harmonisches Mittel: " << harm << endl;
17     return 0;
18 }
```

Kompilieren Sie das Programm und führen Sie es anschließend aus. Kopieren Sie die Ergebnisse als Block-Kommentar an das Ende der Datei `Aufgabe_02_1a.cpp`.

(b) Legen Sie die Datei `Aufgabe_02_1b.cpp` an. Ergänzen Sie darin das folgende C++-Programmfragment so, dass bei der Ausführung des Programms der Variablen `median` der Wert des Medians, und der Variablen `sum` die Summe aller Werte aus `A` zugewiesen werden, die kleiner oder gleich dem Median der durch `A` repräsentierten Folge sind. (1 Punkt)

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      double A[] = { 64.42, 60.27, 14.99, 39.55, 89.02,
6                     49.63, 40.07, 18.16, 16.4, 10.71 };
7      int n = sizeof(A) / sizeof(double);
8      double median;
9      double sum;
10
11     //************************************
12     // Hier beliebig viele Zeilen ergaenzen
13     //************************************
```

```
14
15      cout << "Median: " << median << endl;
16      cout << "Summe: " << sum << endl;
17      return 0;
18  }
```

Kompilieren Sie das Programm und führen Sie es anschließend aus. Kopieren Sie die Ergebnisse als Block-Kommentar an das Ende der Datei `Aufgabe_02_1b.cpp`.

(c) Legen Sie die Datei `Aufgabe_02_1c.cpp` an. Ergänzen Sie darin das C++-Programmfragment aus Aufgabenteil (b) so, dass bei der Ausführung des Programms der Variablen `sum` die Summe $a_{(1)} + a_{(3)} + \ldots$ aller Elemente aus `A` mit ungeradem Rang zugewiesen wird. Der Median spielt für diesen Aufgabenteil keine Rolle. (1 Punkt)

Kompilieren Sie das Programm und führen Sie es anschließend aus. Kopieren Sie die Ergebnisse als Block-Kommentar an das Ende der Datei `Aufgabe_02_1c.cpp`.

Hinweis: Die Programme zu (a), (b) und (c) sollen auch dann noch die korrekten Werte berechnen, wenn dem Array `A` Werte hinzugefügt oder Werte daraus entfernt werden, ohne dass weitere Änderungen am Programm vorgenommen werden müssen. Eine Berechnung per Hand mit anschließender direkter Initialisierung der Variablen `harm`, `median` bzw. `sum` ist also nicht zulässig. Für einen Teil der Lösungen von (b) und (c) empfiehlt es sich nochmals Blatt 0 anzuschauen. Sie dürfen nicht die Bibliotheken `cmath` und `math.h` verwenden.

(Präsenz-)Aufgabe 2: Reihenfolge (0 Punkte)

Ergänzen Sie das folgende C++-Programmfragment so, dass bei Ausführung die Zahlen des Arrays `feld` ausgegeben werden. Wenn eine Zahl sich *unmittelbar* (gegebenenfalls mehrfach) wiederholt, soll nur das erste Vorkommen ausgegeben werden.

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int feld[] = { 8, 3, 3, 38, 1, 1, 1, 9,
6                     4, 7, 8, 8, 3, 50, 50, 9 };
7      int n = sizeof(feld) / sizeof(int);
8
9      cout << "Array ohne doppelte Vorkommen: " << endl;
10
11     //************************************
12     // Hier beliebig viele Zeilen ergaenzen
13     //************************************
14
15     return 0;
16 }
```

Beispielausgabe:

```
Array ohne doppelte Vorkommen:
8 3 38 1 9 4 7 8 3 50 9
```