

Freiwillige Offline-Aufgabe 02-04 (INF & WI & MCD): Zeit-Berechnungen mit Ganzzahlen

Die Lösung dieser Aufgabe kann unter Benutzung des „modulo“ Operators programmiert werden. Siehe Übung.

Teilaufgabe a: (Eingaben, Ganzzahlrechnung)

Schreiben Sie ein C++ Programm, welches eine Uhrzeit einliest. Dabei sollen die Stunden, Minuten und Sekunden der Uhrzeit getrennt eingelesen werden. Die Stunden nach Mittag sollen in „24-Stunden-Notation“ eingegeben werden.

Das Programm soll dann für die eingegebene Uhrzeit die Anzahl der Sekunden seit Mitternacht berechnen und ausgeben.

Ihr Programm soll annehmen, dass der Benutzer nur korrekte Eingaben macht. Zahlen kleiner als 10 werden einstellig ausgegeben, was nicht schön ist, hier aber vernachlässigt werden soll.

Testläufe (Benutzereingaben unterstrichen):

Bitte geben Sie die Stundenzahl der ersten Uhrzeit ein: ? 1
Bitte geben Sie die Minutenzahl der ersten Uhrzeit ein: ? 11
Bitte geben Sie die Sekundenzahl der ersten Uhrzeit ein: ? 11
1:11 Uhr und 11 Sekunden sind in Sekunden seit Mitternacht: 4271
Drücken Sie eine beliebige Taste . . .

Bitte geben Sie die Stundenzahl der ersten Uhrzeit ein: ? 2
Bitte geben Sie die Minutenzahl der ersten Uhrzeit ein: ? 2
Bitte geben Sie die Sekundenzahl der ersten Uhrzeit ein: ? 22
2:2 Uhr und 22 Sekunden sind in Sekunden seit Mitternacht: 7342
Drücken Sie eine beliebige Taste . . .

Teilaufgabe b: (Eingabe und Berechnung für eine zweite Uhrzeit)

Erweitern Sie das C++ Programm aus der vorigen Teilaufgabe um eine identische Eingabe und Berechnung für eine zweite Uhrzeit.

Testlauf (Benutzereingaben unterstrichen):

Bitte geben Sie die Stundenzahl der ersten Uhrzeit ein: ? 1
Bitte geben Sie die Minutenzahl der ersten Uhrzeit ein: ? 11
Bitte geben Sie die Sekundenzahl der ersten Uhrzeit ein: ? 11
1:11 Uhr und 11 Sekunden sind in Sekunden seit Mitternacht: 4271

Bitte geben Sie die Stundenzahl der zweiten Uhrzeit ein: ? 4
Bitte geben Sie die Minutenzahl der zweiten Uhrzeit ein: ? 44
Bitte geben Sie die Sekundenzahl der zweiten Uhrzeit ein: ? 0
4:44 Uhr und 0 Sekunden sind in Sekunden seit Mitternacht: 17040

Drücken Sie eine beliebige Taste . . .

Teilaufgabe c: (Berechnung und Ausgabe der Zeitdifferenz)

Erweitern Sie das C++ Programm aus der vorigen Teilaufgabe um die Berechnung und Ausgabe der Zeitdifferenz (in Stunden, Minuten, Sekunden) zwischen den beiden eingegebenen Uhrzeiten.

Die zweite eingegebene Uhrzeit sei dabei immer „später am Tag“ als die erste Uhrzeit.

Testlauf (Benutzereingaben unterstrichen):

Bitte geben Sie die Stundenzahl der ersten Uhrzeit ein: ? 3
Bitte geben Sie die Minutenzahl der ersten Uhrzeit ein: ? 33
Bitte geben Sie die Sekundenzahl der ersten Uhrzeit ein: ? 33
3:33 Uhr und 33 Sekunden sind in Sekunden seit Mitternacht: 12813

Bitte geben Sie die Stundenzahl der zweiten Uhrzeit ein: ? 6
Bitte geben Sie die Minutenzahl der zweiten Uhrzeit ein: ? 5
Bitte geben Sie die Sekundenzahl der zweiten Uhrzeit ein: ? 4
6:5 Uhr und 4 Sekunden sind in Sekunden seit Mitternacht: 21904

Die Differenz zwischen den beiden Uhrzeiten betraegt:

2 Stunden, 31 Minuten und 31 Sekunden.

Drücken Sie eine beliebige Taste . . .

GIP-INF, GIP-WI/MCD, WS 2020/2021

Freiwillige Offline-Aufgabe 02-04 (INF & WI & MCD)

Prof. Dr. Andreas Claßen

Bitte geben Sie die Stundenzahl der ersten Uhrzeit ein: ? 0
Bitte geben Sie die Minutenzahl der ersten Uhrzeit ein: ? 0
Bitte geben Sie die Sekundenzahl der ersten Uhrzeit ein: ? 0
0:0 Uhr und 0 Sekunden sind in Sekunden seit Mitternacht: 0

Bitte geben Sie die Stundenzahl der zweiten Uhrzeit ein: ? 17
Bitte geben Sie die Minutenzahl der zweiten Uhrzeit ein: ? 20
Bitte geben Sie die Sekundenzahl der zweiten Uhrzeit ein: ? 35
17:20 Uhr und 35 Sekunden sind in Sekunden seit Mitternacht: 62435

Die Differenz zwischen den beiden Uhrzeiten betraegt:
17 Stunden, 20 Minuten und 35 Sekunden.
Drücken Sie eine beliebige Taste . . .

Freiwillige Offline-Aufgabe O 02-05 (INF & WI & MCD):

Kleinbuchstaben in Großbuchstaben umwandeln (Berechnungen mit Ganzzahlen, ASCII Tabelle)

Aufgabe INF-WI-MCD-O-02.05: (Kleinbuchstaben in Großbuchstaben umwandeln)

Schreiben Sie ein C++ Programm, welches einen Kleinbuchstaben über die Tastatur einliest und den entsprechenden Großbuchstaben ausgibt. *Sie können davon ausgehen, dass der Benutzer nur korrekte Eingaben macht.*

Hinweis: Berechnen Sie die Position im Alphabet des Kleinbuchstabens (siehe Praktikumsaufgabe). Geben Sie dann aus dem Alphabet der Großbuchstaben den Buchstaben an der identischen Position aus.

Testläufe (Benutzereingaben sind zur Verdeutlichung unterstrichen):

Bitte geben Sie den Kleinbuchstaben ein: ? a
Der entsprechende Grossbuchstabe lautet: A
Drücken Sie eine beliebige Taste . . .

Bitte geben Sie den Kleinbuchstaben ein: ? d
Der entsprechende Grossbuchstabe lautet: D
Drücken Sie eine beliebige Taste . . .

Bitte geben Sie den Kleinbuchstaben ein: ? z
Der entsprechende Grossbuchstabe lautet: Z
Drücken Sie eine beliebige Taste . . .

Die Unterstreichung in den Testläufen dient nur zur Erläuterung des Testlaufs (d.h. um klar zu machen, welche Teile des Testlaufs eine Benutzereingabe sind). Sie soll bei Ihnen auf dem Bildschirm nicht sichtbar sein.

Pflicht-Offline-Aufgabe O 01-01, INF & WI & MCD:

Erstellen Sie ein C++ Programm, das den Text `Hello Jenkins!` gefolgt von einem Zeilenumbruch ausgibt.

Testlauf (auf Ihrem Rechner) des Programms:

`Hello Jenkins!`
Drücken Sie eine beliebige Taste . . .

Entwickeln und testen Sie das Programm auf Ihrem Rechner und laden Sie es als Lösung der Offline-Aufgabe *O 01-01* ins Jenkins System hoch.
Stellen Sie sicher, dass Jenkins Ihr Programm akzeptiert („blaue Kugel“).

Pflicht-Offline-Aufgabe 02-01, INF & MCD & WI:

Berechnung des Umfangs und der Fläche eines Quadrats

Schreiben Sie ein C++ Programm, welches die Seitenlänge eines Quadrats als Fließkommazahl einliest (in cm) und den Umfang (in cm) sowie die Fläche (in cm^2) des Quadrats berechnet und ausgibt.

Bei der Eingabeaufforderung soll ihr Programm jeweils die Textzeile „Bitte geben Sie ...“ gefolgt von einem Doppelpunkt gefolgt von einem Leerzeichen gefolgt von einem Fragezeichen gefolgt von einem Leerzeichen ausgeben. D.h. genau diese Ausgabe müssen Sie in Ihrem Programm programmieren.

Der Benutzer mache nur korrekte Eingaben. D.h. der Benutzer gibt keine negativen Zahlen ein und macht auch keine Eingaben, die keine Fließkommazahl sind.

Wird vom Benutzer eine ganze Zahl eingegeben dort, wo eine Fließkommazahl erwartet wird, so wird diese automatisch umgewandelt.

Testläufe (Benutzereingaben unterstrichen):

Bitte geben Sie die Seitenlaenge ein (in cm): ? 5.0
Der Umfang des Quadrats betraegt (in cm): 20
Die Flaeche des Quadrats betraegt (in cm^2): 25
Drücken Sie eine beliebige Taste . . .

Bitte geben Sie die Seitenlaenge ein (in cm): ? 6.7
Der Umfang des Quadrats betraegt (in cm): 26.8
Die Flaeche des Quadrats betraegt (in cm^2): 44.89
Drücken Sie eine beliebige Taste . . .

Bitte geben Sie die Seitenlaenge ein (in cm): ? 0
Der Umfang des Quadrats betraegt (in cm): 0
Die Flaeche des Quadrats betraegt (in cm^2): 0
Drücken Sie eine beliebige Taste . . .

Pflicht-Offline-Aufgabe 02-02, INF & WI & MCD:

Tausch der Werte zweier Variablen mittels Ringtausch / Dreieckstausch

Gegeben seien zwei C++ Variablen `v1` und `v2` des gleichen Datentyps, z.B. `int`. Beispiel:

```
int v1 = 4;  
int v2 = 7;
```

Möchte man die Werte, die gerade in den beiden Variablen gespeichert sind, tauschen (so dass im obigen Beispiel die Variable `v1` nachher den Wert 7 und die Variable `v2` nachher den Wert 4 enthält), so ist Vorsicht geboten. Bei zwei Wertzuweisungen ...

```
v1 = v2;  
v2 = v1;
```

... würde durch die erste Wertzuweisung der bisherige Wert 4 der Variable `v1` durch den Wert 7 der Variable `v2` überschrieben. Nach dieser ersten Wertzuweisung wäre somit in beiden Variablen der Wert 7 gespeichert. Der Wert 4 wäre verloren. Dies kann auch durch die zweite Wertzuweisung dann nicht mehr „repariert“ werden, in beiden Variablen wäre nachher der Wert 7 gespeichert.

Genauso würde umgekehrt durch die Wertzuweisungen ...

```
v2 = v1;  
v1 = v2;
```

... der Wert 7 verloren gehen.

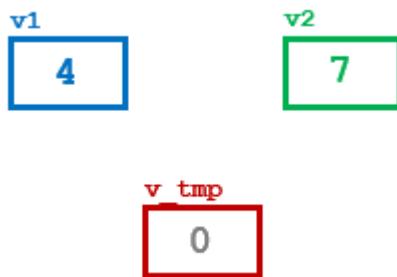
Als „bildliche Analogie“: Zwei Personen, jeder hält ein Paket in der Hand und kann auch nur ein Paket halten. Wie können die beiden ihre Pakete tauschen („gleichzeitiges Zuwerfen“ geht nicht)? Am einfachsten mittels einer dritten Person, die assistiert ...

(Der Vergleich stimmt im Detail nicht ganz überein, aber das soll hier egal sein ...)



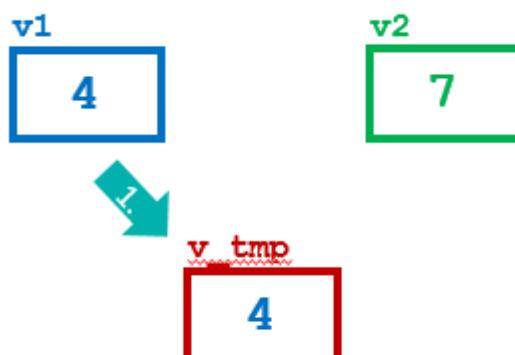
Aus diesem Grund wird der Tausch der Werte zweier Variablen meist dadurch realisiert, dass man eine dritte Variable des gleichen Datentyps anlegt, die nur temporär (kurzzeitig) „als Assistent“ für den Wertetausch benötigt wird:

```
int v_tmp = 0;
```



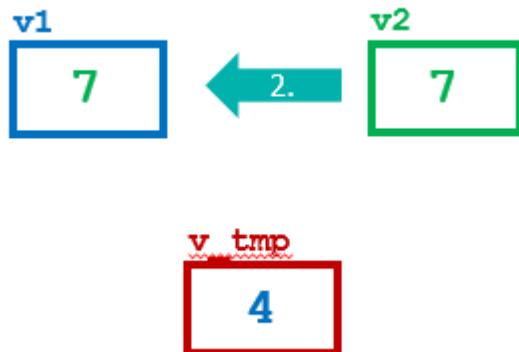
Dann werden die Werte von v1 und v2 dadurch getauscht, dass zuerst einer der beiden Werte (z.B. der von v1) in der neuen Variable gespeichert wird:

```
v_tmp = v1;
```



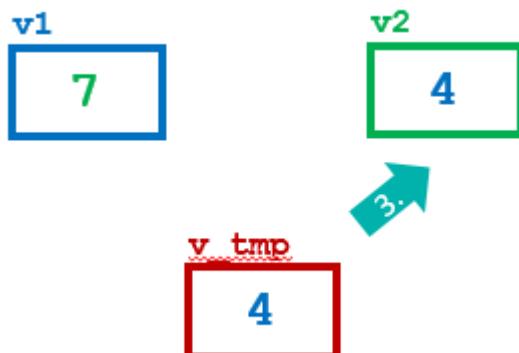
Dann kann dieser Wert von `v1` mit dem Wert von `v2` überschrieben werden, da der Wert ja in `v_tmp` „gesichert“ worden ist:

`v1 = v2;`



Dann kann der Wert von `v2` letztlich mit dem Wert von `v_tmp` (d.h. mit dem „gesicherten ursprünglichen Wert von `v1`“) überschrieben werden:

`v2 = v_tmp;`



Damit ist das Ziel des Wertetauschs zwischen v1 und v2 erreicht. v_tmp und sein Wert werden nicht weiter benutzt.



Schreiben Sie ein C++ Programm, welches zwei ganze Zahlen einliest und in zwei Variablen speichert und dann mittels einer von Ihnen im Programm angelegten dritten Variablen und des beschriebenen Ringtausch-Verfahrens die Werte der beiden ursprünglichen Variablen tauscht und wieder ausgibt.

Der Benutzer mache nur korrekte Eingaben, d.h. gebe nur korrekte ganze Zahlen (positiv, negativ oder Null) ein.

Testläufe (Benutzereingaben unterstrichen):

Bitte geben Sie den ganzzahligen Wert der ersten Variable ein: ? 4

Bitte geben Sie den ganzzahligen Wert der zweiten Variable ein: ? 7

Nach dem Tausch:

Wert der ersten Variable: 7

Wert der zweiten Variable: 4

Drücken Sie eine beliebige Taste . . .

Bitte geben Sie den ganzzahligen Wert der ersten Variable ein: ? -35

Bitte geben Sie den ganzzahligen Wert der zweiten Variable ein: ? 99

Nach dem Tausch:

Wert der ersten Variable: 99

Wert der zweiten Variable: -35

Drücken Sie eine beliebige Taste . . .

Bitte geben Sie den ganzzahligen Wert der ersten Variable ein: ? 5

Bitte geben Sie den ganzzahligen Wert der zweiten Variable ein: ? 5

Nach dem Tausch:

Wert der ersten Variable: 5

Wert der zweiten Variable: 5

Drücken Sie eine beliebige Taste . . .

Bitte geben Sie den ganzzahligen Wert der ersten Variable ein: ? 0

Bitte geben Sie den ganzzahligen Wert der zweiten Variable ein: ? 0

Nach dem Tausch:

Wert der ersten Variable: 0

Wert der zweiten Variable: 0

Drücken Sie eine beliebige Taste . . .

Freiwillige Offline-Aufgabe 02-03, INF & WI & MCD:

Berechnungen mit Ganzzahlen und Fließkommazahlen

*Es soll nur die Lösung der Teilaufgabe b) ins System hochgeladen werden.
Grund: Die Lösung von b) ist eine Erweiterung der Lösung von a).*

Teilaufgabe a: (Eingaben, Ganzzahlrechnung)

Schreiben Sie ein C++ Programm, welches zu einer politischen Wahl mit drei Kandidaten die Gesamtzahl der abgegebenen gültigen Stimmen als Ganzzahl einliest. Anschließend sollen die Anzahlen der Stimmen eingelesen werden, die auf die ersten beiden Kandidaten entfallen sind.

Das Programm soll dann die Zahl der verbleibenden Stimmen berechnen und als Zahl der Stimmen für den dritten Kandidaten ausgeben (*auch wenn diese Art der Stimmermittlung nicht realistisch ist*).

Der Benutzer soll nur korrekte Eingaben machen.

Testläufe (Benutzereingaben unterstrichen):

Bitte geben Sie die Gesamtzahl der abgegebenen gültigen Stimmen ein: ? 10000

Bitte geben Sie die Anzahl Stimmen des ersten Kandidaten ein: ? 2222

Bitte geben Sie die Anzahl Stimmen des zweiten Kandidaten ein: ? 3333

Auf den dritten Kandidaten sind somit 4445 Stimmen entfallen.

Drücken Sie eine beliebige Taste . . .

Bitte geben Sie die Gesamtzahl der abgegebenen gültigen Stimmen ein: ? 1234

Bitte geben Sie die Anzahl Stimmen des ersten Kandidaten ein: ? 123

Bitte geben Sie die Anzahl Stimmen des zweiten Kandidaten ein: ? 234

Auf den dritten Kandidaten sind somit 877 Stimmen entfallen.

Drücken Sie eine beliebige Taste . . .

Bitte geben Sie die Gesamtzahl der abgegebenen gültigen Stimmen ein: ? 100

Bitte geben Sie die Anzahl Stimmen des ersten Kandidaten ein: ? 0

Bitte geben Sie die Anzahl Stimmen des zweiten Kandidaten ein: ? 33

Auf den dritten Kandidaten sind somit 67 Stimmen entfallen.

Drücken Sie eine beliebige Taste . . .

Teilaufgabe b: (Berechnung der Stimmanteile in Prozent)

Erweitern Sie das C++ Programm aus der vorigen Teilaufgabe: Es sollen nun auch die Stimmanteile in Prozent, berechnet aus den einzelnen Stimmanzahlen und der Gesamtanzahl an Stimmen, ausgegeben werden.

Testläufe (Benutzereingaben unterstrichen):

Bitte geben Sie die Gesamtzahl der abgegebenen gueltigen Stimmen ein: ? 10000

Bitte geben Sie die Anzahl Stimmen des ersten Kandidaten ein: ? 2222

Bitte geben Sie die Anzahl Stimmen des zweiten Kandidaten ein: ? 3333

Auf den dritten Kandidaten sind somit 4445 Stimmen entfallen.

Kandidat 1 erhielt 22.22% der Stimmen.

Kandidat 2 erhielt 33.33% der Stimmen.

Kandidat 3 erhielt 44.45% der Stimmen.

Drücken Sie eine beliebige Taste . . .

Bitte geben Sie die Gesamtzahl der abgegebenen gueltigen Stimmen ein: ? 1234

Bitte geben Sie die Anzahl Stimmen des ersten Kandidaten ein: ? 123

Bitte geben Sie die Anzahl Stimmen des zweiten Kandidaten ein: ? 234

Auf den dritten Kandidaten sind somit 877 Stimmen entfallen.

Kandidat 1 erhielt 9.96759% der Stimmen.

Kandidat 2 erhielt 18.9627% der Stimmen.

Kandidat 3 erhielt 71.0697% der Stimmen.

Drücken Sie eine beliebige Taste . . .

Bitte geben Sie die Gesamtzahl der abgegebenen gueltigen Stimmen ein: ? 100

Bitte geben Sie die Anzahl Stimmen des ersten Kandidaten ein: ? 0

Bitte geben Sie die Anzahl Stimmen des zweiten Kandidaten ein: ? 33

Auf den dritten Kandidaten sind somit 67 Stimmen entfallen.

Kandidat 1 erhielt 0% der Stimmen.

Kandidat 2 erhielt 33% der Stimmen.

Kandidat 3 erhielt 67% der Stimmen.

Drücken Sie eine beliebige Taste . . .
