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

Zieldatum (nicht bindend): Mi 24.10.2018

2018-10-19: Fehler korrigiert: Testläufe passten nicht zur Aufgabe.

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;
```

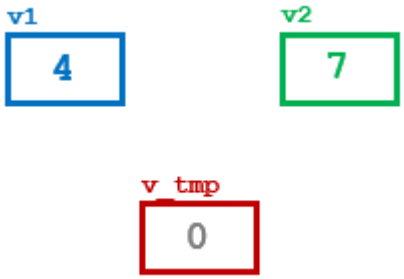
… der Wert `7` verloren gehen.

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 …)

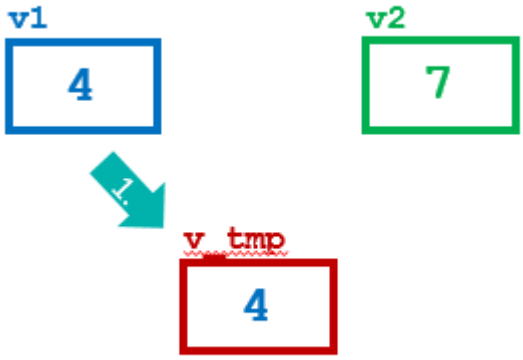
Aus diesem Grund wird der Tausch der Werte zweier Variablen meist dadurch realisiert, dass man eine dritte Variable des gleichen Datentyps anlegt, die nur temporär (kurzzeitig) „als Assistent" für den Wertetausch benötigt wird:

```
int v_tmp = 0;
```



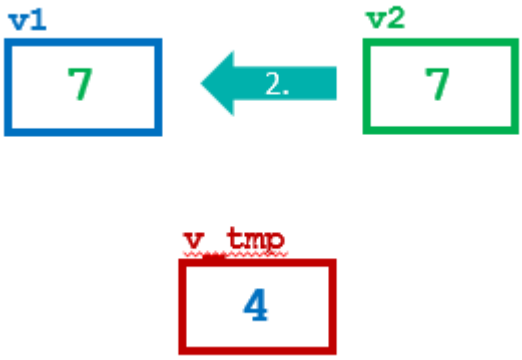
Dann werden die Werte von `v1` und `v2` dadurch getauscht, dass zuerst einer der beiden Werte (z.B. der von `v1`) in der neuen Variable gespeichert wird:

```
v_tmp = v1;
```



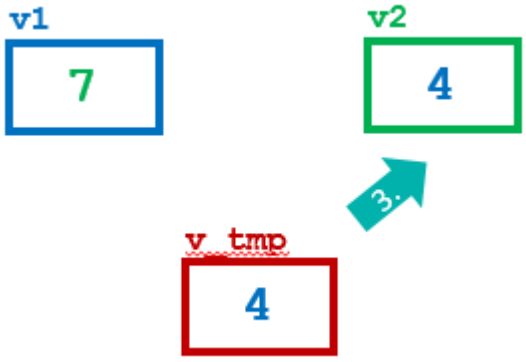
Dann kann dieser Wert von `v1` mit dem Wert von `v2` überschrieben werden, da der Wert ja in `v_tmp` „gesichert" worden ist:

```
v1 = v2;
```

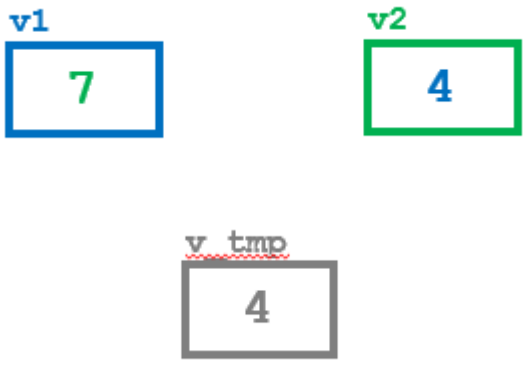


Dann kann der Wert von `v2` letztlich mit dem Wert von `v_tmp` (d.h. mit dem „gesicherten ursprünglichen Wert von `v1`") überschrieben werden:

```
v1 = 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 . . .
```