

Prof. Markus Gross, Bruno Heidelberger, Richard Keiser, Nicky Kern, Edouard Lamboray, Christoph Niederberger, Tim Weyrich, Felix Eberhard, Manuel Graber, Nathalie Kellenberger, Marcel Kessler, Lior Wehrli

Uebung 3- Logische Ausdrücke / Verzweigung

Ausgabe: 17. November 2003
Abgabe: 24. November 2003
Autor: Edouard Lamboray

1. Logische & relationale Ausdrücke

3 Punkte

C++ kennt folgende Operatoren der booleschen Logik: || (OR), && (AND) und ! (NOT).

Formuliere nun anhand dieser Operatoren die folgenden Ausdrücke:

Beispiel: a ist kleiner als 3 und nicht kleiner oder gleich 0

Antwort: $(a < 3) \ \&\& \ ! (a \leq 0)$ oder aber auch $(a < 3) \ \&\& \ (a > 0)$

- a) x ist eine gerade (natürliche) Zahl, welche aber nicht im Bereich $]13,17]$ ¹ liegt.
- b) Die absolute Differenz zwischen x und y ist grösser gleich 30 und {x liegt über 100 oder y ist kleiner gleich 200}.
- c) ch ist ein char, der in [a,z] oder [A,Z] liegt (also ein Buchstabe ist)².

Formuliere nun folgende Ausdrücke umgangssprachlich. Beachte dabei die Klammern und die Reihenfolge der Auswertung:

- d) $a > b \ || \ a \neq c \ \&\& \ b > c$
- e) $(ch == 'q' \ || \ ch == 'Q') \ \&\& \ mouseButtonState \ != \ pressed$
- f) $x - y > 0 \ \&\& \ x \ != \ y \ \&\& \ (x + y \leq 0 \ || \ x == y)$

2. Logische bitweise Operatoren

4 Punkte

Im Gegensatz zu den booleschen Operatoren gibt es in C/C++ noch die logischen bitweisen Operatoren. Diese arbeiten nicht auf den Variablenwerten, sondern beziehen sich auf die Bits der internen Repräsentation einer Variablen.

Im folgenden werden die Operatoren an Variablen vom Typ unsigned char erklärt. Wir stellen alle Werte binär dar. Ein Wert W ergibt sich aus seiner Binärdarstellung auf N Bits durch

$$W = \sum_{i=0}^{N-1} b_i \cdot 2^i$$

-
- 1. $]a,b]$ beinhaltet alle Zahlen von (aber ohne) a bis (und mit) b.
 - 2. Gehe davon aus, dass Buchstaben von a bis z und von A bis Z jeweils aufeinanderfolgend sind.

Als Beispiel sind nun sechs Werte und ihre Binärdarstellung aufgeführt, wobei die Bits von rechts nach links gelesen werden:

0	0 0 0 0 0 0 0 0	3	0 0 0 0 0 0 1 1
1	0 0 0 0 0 0 0 1	100	0 1 1 0 0 1 0 0
2	0 0 0 0 0 0 1 0	255	1 1 1 1 1 1 1 1

Die logischen bitweisen Operatoren operieren auf allen Bits der Variable gleichzeitig, indem sie den Operator jeweils auf Bits an derselben Stelle anwenden:

AND-Operator: $a \& b$

a	0 1 1 0 0 1 0 1
b	1 1 0 0 0 0 1 1
$a \& b$	0 1 0 0 0 0 0 1

OR-Operator: $a | b$

a	0 1 1 0 0 1 0 1
b	1 1 0 0 0 0 1 1
$a b$	1 1 1 0 0 1 1 1

XOR-Operator: $a \wedge b$

a	0 1 1 0 0 1 0 1
b	1 1 0 0 0 0 1 1
$a \wedge b$	1 0 1 0 0 1 1 0

NOT-Operator: $\sim a$

a	0 1 1 0 0 1 0 1
$\sim a$	1 0 0 1 1 0 1 0

Shift-Operator: $a \ll n$

Der Shift-Operator schiebt die Bits um n Stellen nach links und fügt von rechts jeweils eine 0 ein:

a	0 1 1 0 0 1 0 1
$a \ll 1$	1 1 0 0 1 0 1 0
$a \ll 4$	0 1 0 1 0 0 0 0

Mit diesen Operatoren werden wir nun die Ausgabe eines Wertes in seiner Binärdarstellung implementieren. Dafür verwenden wir folgende Ueberlegung:

- Wir müssen 8 mal eine '1' oder '0' ausgeben
- Wenn wir das Bit an der i-ten Stelle prüfen wollen, müssen wir ein bitweises AND auf die Variable und eine Maske mit einer 1 an der i-ten Stelle anwenden. Alle anderen Bits der Maske sind gleich Null (z.B. Maske = 00001000 für $i = 4$). Die Maske kann jeweils mit dem Shift-Operator erstellt werden.

- Mit einem einfachen Test auf das Resultat vom letzten Schritt kann man ganz einfach zwischen einer 0 und einer 1 an der entsprechenden Stelle unterscheiden und dementsprechend eine '0' oder '1' ausgeben.

Implementiere nun die Ausgabe des Binärwertes einer Variable vom Typ `unsigned char`. Das Hauptprogramm soll zuerst den Benutzer nach einer Zahl fragen und diese dann als Binärzahl wieder ausgeben. Erstelle dafür eine Funktion, welche das *i*-te Binärzeichen einer Zahl ausgibt und eine andere Funktion, welche diese Funktion für eine Zahl acht mal aufruft und damit alle acht Binärzeichen ausgibt.

Freiwillig: Fortgeschrittene können die Aufgabe natürlich eleganter mit `for`- oder `while`-Schleife lösen. Dies ist aber nicht notwendig.

3. Die `switch`-Anweisung

3 Punkte

Eine `if`-Anweisung erlaubt zunächst nur die Unterscheidung zwischen zwei Fällen. Erst durch Verschachtelung können mehrere Fälle behandelt werden. Das kann umständlich und unübersichtlich werden. Bei einer Auswahl unter vielen Anweisungen ist die `switch`-Anweisung übersichtlicher.

Aufgabe:

Ein Versandhaus hat fünf verschiedene Produkte im Angebot. Die Preise für diese Produkte sind: 1 – 1.90 CHF, 2 – 4.50 CHF, 3 – 9.95 CHF, 4 – 15.50 CHF und 5 – 27.30 CHF. Die Bestellungen gehen per E-Mail ein und enthalten die Produktnummern und die Bestellmengen pro Produkt, also zum Beispiel 1:0, 2:2, 3:1, 4:2, 5:0. Schreibe ein Programm, welches diese Werte einliest indem es für jede Kategorie nach der Bestellmenge fragt.

Das Programm soll mit Hilfe der `switch`-Anweisung den Preis des jeweiligen Produktes ermitteln. Anschliessend soll der zu bezahlende Preis pro Kunde berechnet und ausgegeben werden.

Achtung: Was passiert, wenn eine negative Kategorie/Menge eingegeben wird?

Eine Ein- und Ausgabe könnte z.B. so aussehen:

```
Bestellmenge Produkt 1?
```

```
0
```

```
Bestellmenge Produkt 2?
```

```
2
```

```
Bestellmenge Produkt 3?
```

```
1
```

```
Bestellmenge Produkt 4?
```

```
2
```

```
Bestellmenge Produkt 5?
```

```
0
```

```
Der Kunde muss 49.95 CHF bezahlen.
```

4. Drucken unter UNIX (fakultativ)

VPP

An der ETH, und nur an der ETH, wird das vpp-System zum Drucken verwendet. Dafür dient das Programm `vpp`. Informationen über das Programm findest Du in der man-page und unter <http://www.vpp.ethz.ch>. Dort findet sich auch eine Liste der verfügbaren VPP-Stationen und der jeweiligen Drucker.

Finde heraus, wie man ein Ao-Poster auf Folie im RZSPEZ drucken kann.

Auf dem tardis-cluster gibt es noch eine zweite Möglichkeit, zu drucken. Mit dem Kommando `lp -dvpp` kann man Dateien an ein graphisches VPP-Frontend schicken. Das Prinzip bleibt aber dasselbe.

Postscript

An das VPP-Kommando darf man nur Postscript-Dateien übergeben. Postscript ist ein spezielles Dateiformat, das für die Ansteuerung von Druckern verwendet wird. Postscript-Dateien sind (ziemlich komplizierte) Textdateien. Schau Dir doch einfach mal eine an (z.B. mit `xemacs`).

Um nun auch Textdateien drucken zu können, kann man das Programm **a2ps** verwenden.

Um Papier zu sparen, kann man jede beliebige Postscript-Datei so umsortieren, dass man auf jede Seite zwei Ursprungsseiten druckt. Das macht das Programm **psnup**.

Um schliesslich zu schauen, ob in einer Postscript-Datei das Richtige drin steht, oder auch einfach, um sie am Bildschirm zu lesen, gibt es das Programm **ghostview/gv**.