

ÜBUNG Schaltjahrberechnung

Zu einer gegebenen Jahreszahl ist zu berechnen, ob es sich um ein Schaltjahr handelt. Die Regel für Schaltjahre lautet:

Ein Schaltjahr liegt dann vor, wenn

- die Jahreszahl durch 4 teilbar ist,
- ausser sie ist durch 100 teilbar.
- Einzige Ausnahme: Ist die Jahreszahl durch 400 teilbar, liegt jedoch trotzdem ein Schaltjahr vor.

Sind die Jahre 1800 und 2000 Schaltjahre ?

Geben Sie einen Ausdruck an, der den log Wert 1 (true) ausgibt, wenn ein Schaltjahr vorliegt.

$$\begin{array}{c} \text{1, wenn durch 4 teilbar} \\ \underbrace{\hspace{10em}} \\ SJ = (year \% 4 == 0) \ \&\& \\ \underbrace{((year \% 100 != 0))} \ \parallel \underbrace{(year \% 400 == 0)} \\ \text{1, wenn nicht durch 100 teilbar} \qquad \text{1, wenn durch 400 teilbar} \end{array}$$

Gregorianischer Kalender

$$1 \text{ Jahr} = 365,24219052 \text{ Tage}$$

$$1 \text{ julianisches Jahr} = 365,25 \text{ Tage (45 v. Chr.)}$$

$$1 \text{ greg. Jahr} = 365,2425 \text{ Tage (1582)}$$

ÜBUNGEN: if-else-Anweisung

Welcher Wert wird ausgegeben? (Achtung: C-Puzzles, ganz schlechter Code)

Aufg. 1

```
int i=0, a=0, b=2;
...
if (i==0)
    a=7;
else
    b=15;
    a=b+1;
printf("%d",a);
```

Aufg. 2

```
int i=0,b=10,a;
...
if (i=0)
    a=10;
else if (i=2){
    b=15;
    a=b+1;
}
else a=0;
printf("%d",a);
```

Aufg. 3

```
int i=1,a=5,b=10;
...
if (i==0);
{
    a=10;
    b=a+1;
}
printf("%d",a);
```

① $a = 3$!

$a = b + 1$ gehört nicht zum else-Zweig !

Einklümmungen werden vom Compiler nicht ausgewertet !

② Es wird "16" ausgegeben !

$\text{if } (i = 0) \rightarrow \text{if } (0) \rightarrow \text{false}$

$\text{else if } (i = 2) \rightarrow \text{else if } (2) \rightarrow \text{true}$

Vergleichsoperator : $==$

③ Es wird "10" ausgegeben

$\text{if } (i = 0) \dots$
if-Zweig \Rightarrow if ohne jede Wirkung

ÜBUNG: if-else-Anweisung - Wahl der richtigen Bedingung

(Achtung: C-Puzzles, ganz schlechter Code)

1. Vereinfachen Sie:

```
if (a){
    if(b)
        if(c) D;
}else
    if(b)
        if(c) E;
    else F;
    else;
```

2. Was wird gedruckt:

```
x = -1;
y = -1;

if (x>0)
    if(y>0)
        print("1");
else
    print("2");
```

①

```
if ( a && b && c ) D;
if ( !a && b && c ) E;
if ( !a && b && !c ) F;
```

Es passiert nur etwas, wenn B wahr ist!

```
if (b) {
    if (a && c) D;
    if (!a && c) E;
    if (!a && !c) F;
}
```

2. Was wird gedruckt:

```
x = -1;  
y = -1;  
  
if (x>0)  
    if(y>0)  
        print("1");  
else  
    print("2");
```

Else gehört immer zum (rückwärtsgehend)
letzten if, welches noch kein else hat.

Anm.: {} gelten als eine Anweisung und
werden daher übersprungen

Es wird nichts gedruckt, weil:

```
;  
if (x > 0) {  
    if (y > 0)  
        printf("1");  
    else  
        printf("2");  
}
```

ÜBUNGEN: Switch-Anweisung

①

=> "Auto"

②

2. Geben Sie zu folgendem Ausdruck eine äquivalente switch-Anweisung an:

```
if ((num<=8) && (num>=5))
    printf("schlecht \n");
else if ((num==3) || (num==4))
    printf("mittel \n");
else if ((num<=2) && (num>0))
    printf("gut \n");
else
    printf("unmoeglich \n");
```

```
switch (num) {
    case 1 :
    case 2 : printf ("gut \n"); break;
    case 3 :
    case 4 : printf ("mittel \n"); break;
    case 5 :
    case 6 :
    case 7 :
    case 8 : printf ("schlecht \n"); break;
    default : printf ("unmöglich \n");
}
```

ÜBUNGEN: for-Schleife

1. Schreiben Sie ein Programm, welches alle Zahlen des "2-aus-5-Code" tabellarisch ausgibt, also:

```
00011
00101
00110
...
11000
```

auf TI-Board

Standard

```
#include <stdio.h>
```

```
...
```

```
int main (void)
```

```
{
    TI-Board
    SPS-Stick
    BaseStickConfig()
    Init_TI_Board();

```

```
int i,j,k,l,m;
```

```
for(i=0; i<=1; i++){
    for(j=0; j<=1; j++){
        for(k=0; k<=1; k++){
            for(l=0; l<=1; l++){
                for(m=0; m<=1; m++){

                    if(i+j+k+l+m == 2)
                        printf("%d%d%d%d%d\n", i,j,k,l,m);

                }
            }
        }
    }
}
```

```
#ifdef FT SUPPORT
FT puts ( "Fertig" );
#endif
```

```
// -- endless loop --
while (1) { }
```

```
}
```

```
#include <stdio.h>
```

```
int main (void)
{
```

wie nebenstehend

```
return 0;
```

```
}
```

Da das TI-Board kein Terminal besitzt,
muss nach dem Starten der Keil-Entwicklungs-
umgebung noch die Terminal-
Software (TI_Terminal) gestartet
werden !