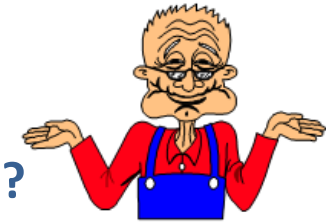
Schleifen in C

Kleine Anekdote ……

1786 kam ein Junge mit 9 Jahren in die Volksschule. Um die Schüler zu beschäftigen, gab der Lehrer die Aufgabe alle Zahlen von 1 bis 100 zusammenzuzählen.

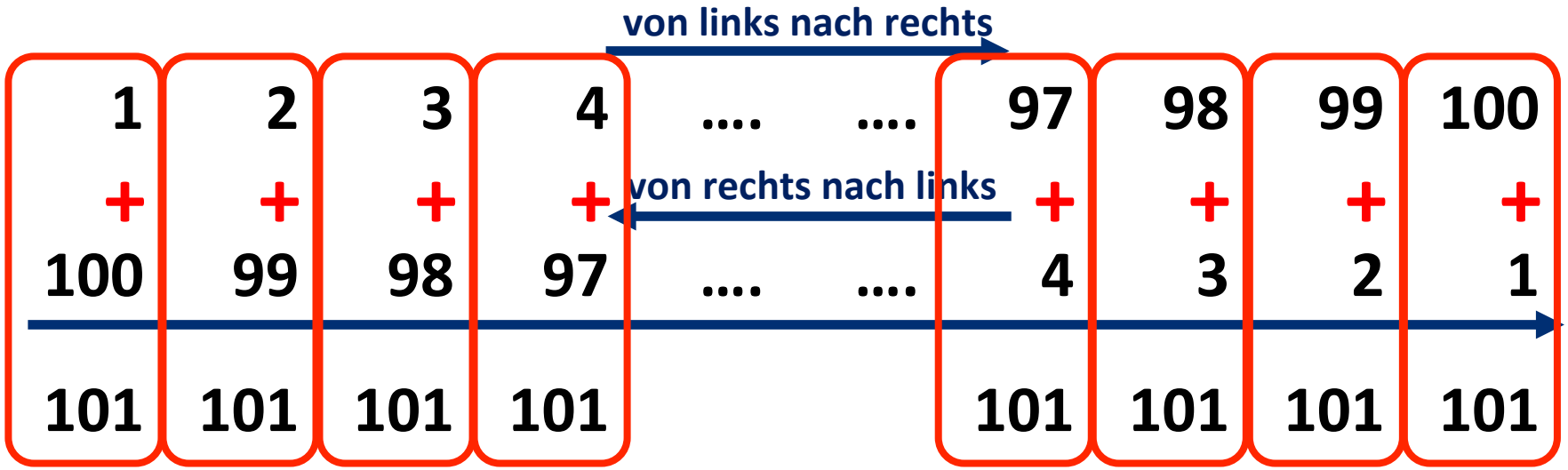
Dieser neunjährige Junge lieferte nach kurzer Zeit das richtige Ergebnis von 5050.

Wie hat er das gemacht?



Versuch der Erklärung ……

Schreibe die Zahlen in zwei Zeilen übereinander.



Es gibt bei 100 Zahlen 50 Zahlenpaare.

Addiere 50 mal die Zahl 101 auf → 5050

wiederhole 50mal den gleichen Vorgang ….

Gaußsche Summenformel

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}.$$

Johann Carl Friedrich Gauß
30. 04. 1777 – 23. 02. 1855

$$\sum_{k=1}^{2} k = 1 + 2 = 3 = \frac{2(2+1)}{2}$$

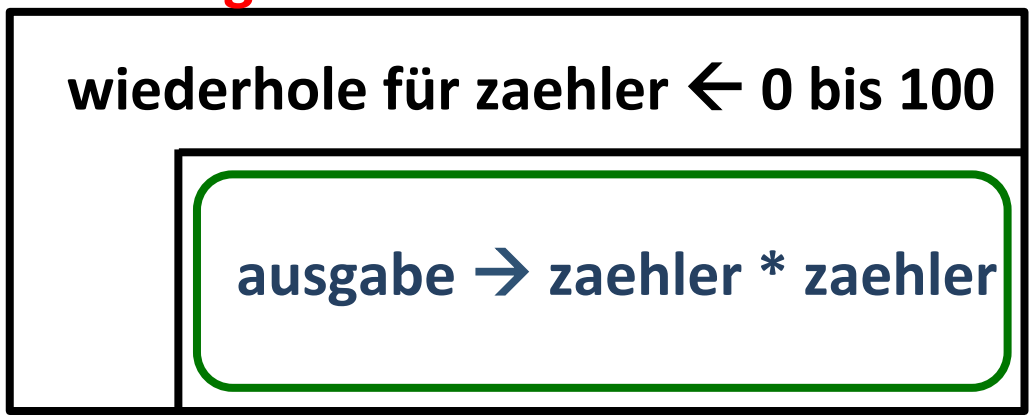
Addiere bis k = 2.

Schleifen

Kontrollstruktur in Programmiersprachen

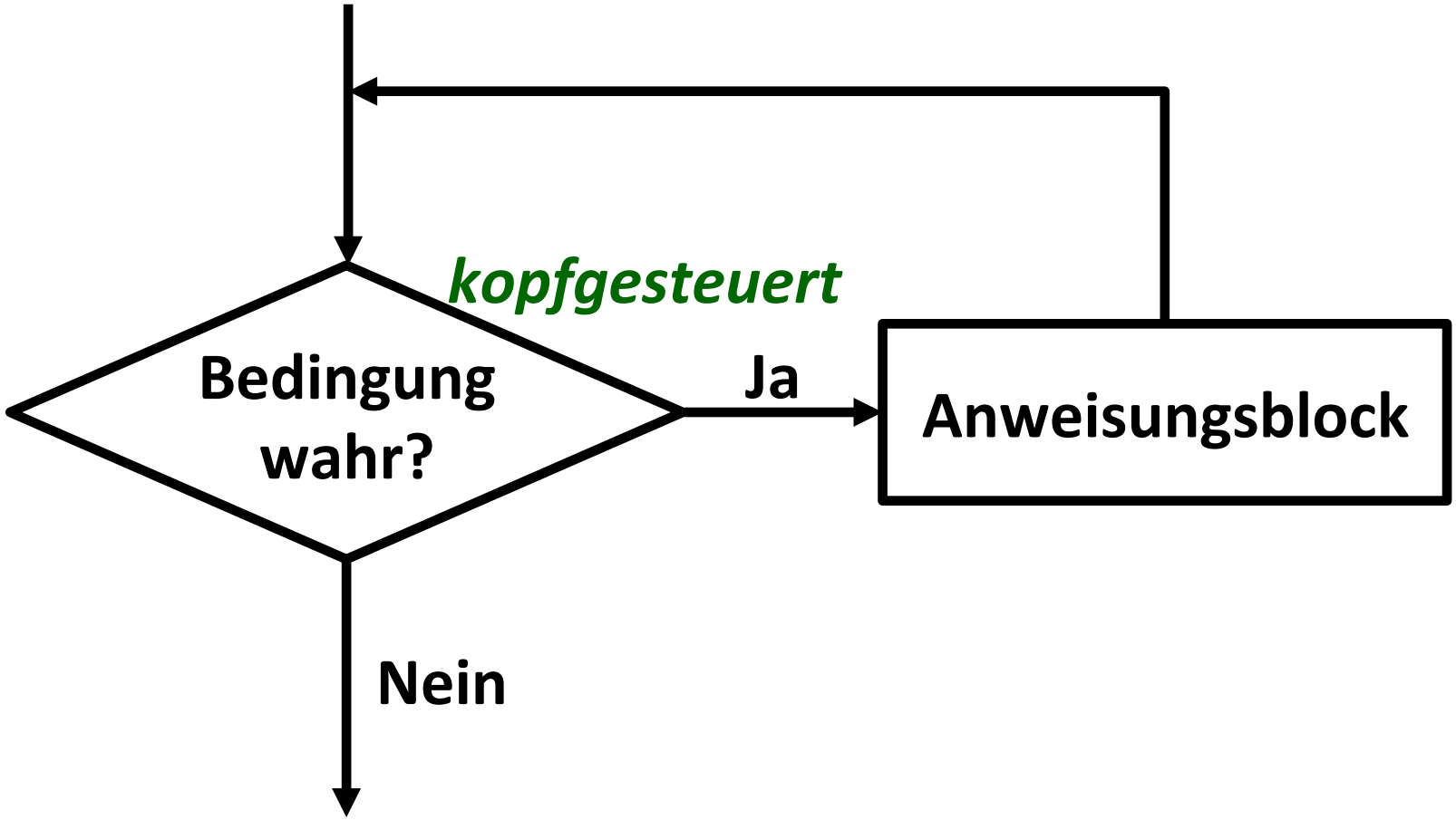
wiederholt einen Code-Block, so lange eine bestimmte Laufzeitbedingung erfüllt ist.

Struktogramm:



C verfügt über die while-, die do … while und die for-Schleife.

while - Schleife



Anweisungsblock wird ausgeführt, so lange die Prüfbedingung true liefert.

while - Schleife *...realisiert im C Quellcode....*

Schlüsselwort

Prüfbedingung für das Ausführen des Anweisungsblocks in der Schleife

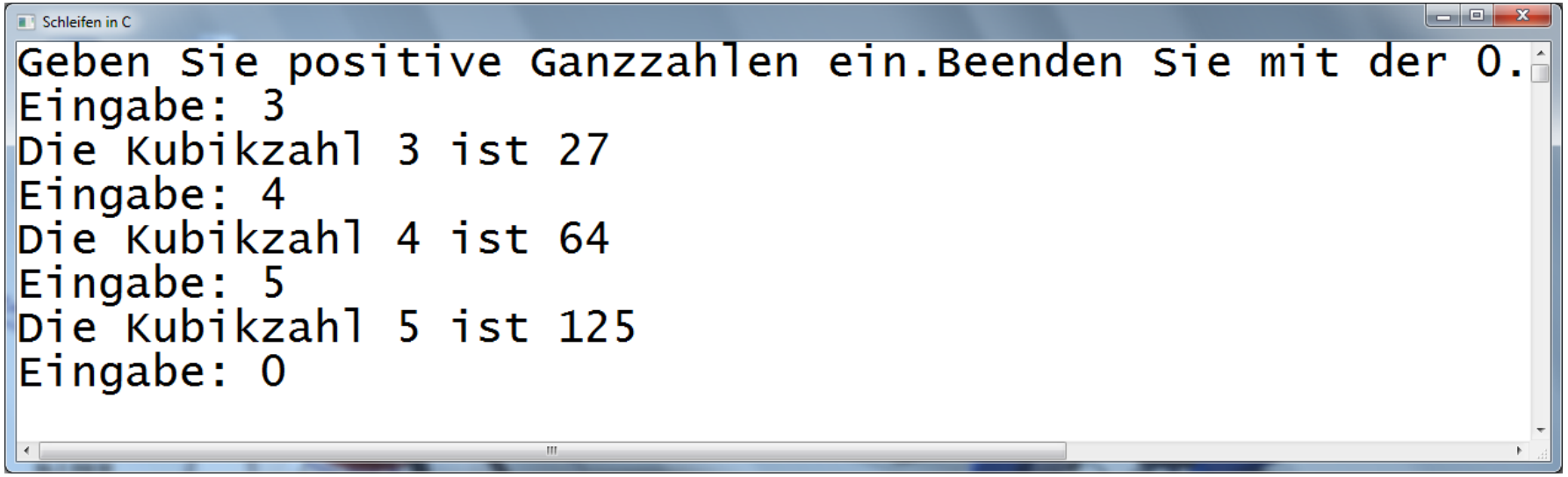
```
while( n > 0 ){
    printf("Die Kubikzahl %d ist %d\n", n, n*n*n);
    printf("Eingabe: ");
    scanf("%d", &n);
}
```

Anweisungsblock

fassen Anweisungsblock in der Schleife zusammen

Abbruchkriterium wäre 0 oder eine negative Zahl.

while - Schleife *...Ausgabe....*

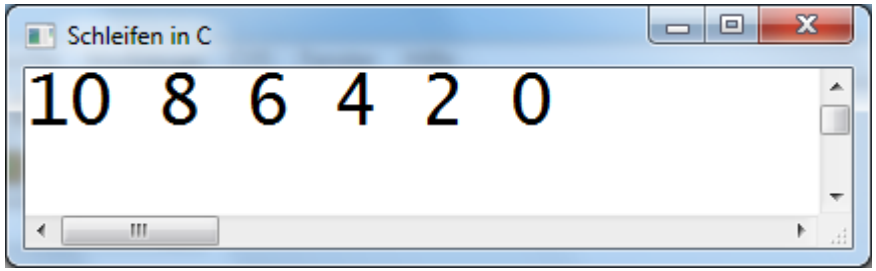



```
while( n > 0 ){

    printf("Die Kubikzahl %d ist %d\n", n, n*n*n);
    printf("Eingabe: ");
    scanf("%d", &n);

}
```

while - Schleife *...kleine Verständnisaufgabe....*

Wie lautet die Ausgabe?

```
int i = -1, j = 11;

while( ++i < j-- ){

        printf("%d ", j-i);
}
```



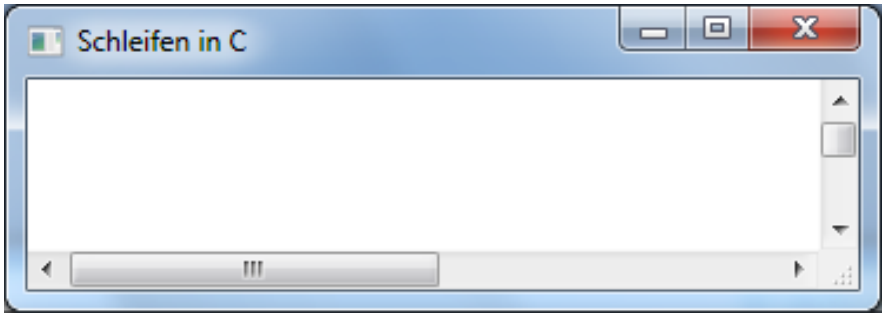
while - Schleife *…noch eine ….*

Wie oft wird die folgende Schleife durchlaufen?

```
int u = 1;

while( u-- ){

        u++;

}
```

Prüfbedingung immer 1 (True)

Endlosschleife

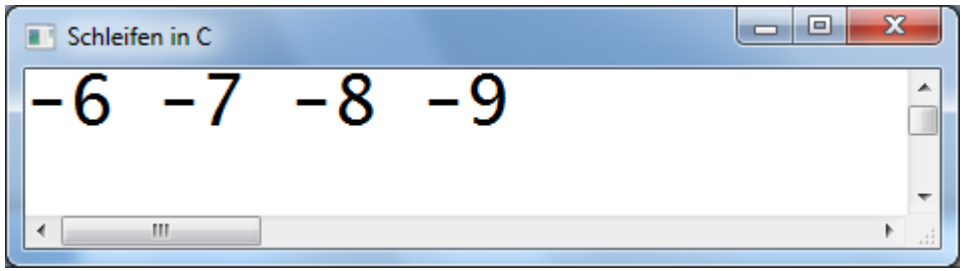


while - Schleife *…noch eine ….*

Wie oft wird die folgende Schleife durchlaufen?

```
int i = 1, j = -5;

while( i+j ){

        printf("%d ", j-i++);
}
```

Ausgabe

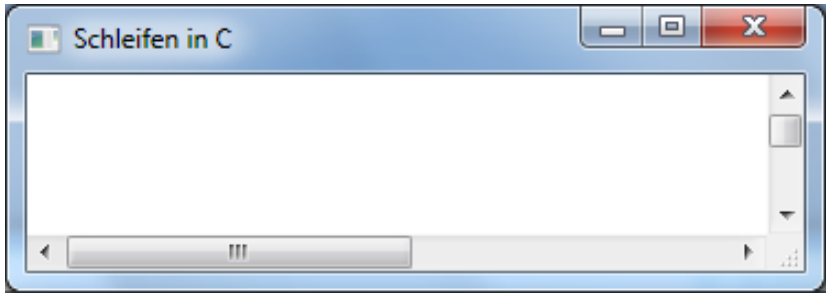


while - Schleife *…noch eine ….*

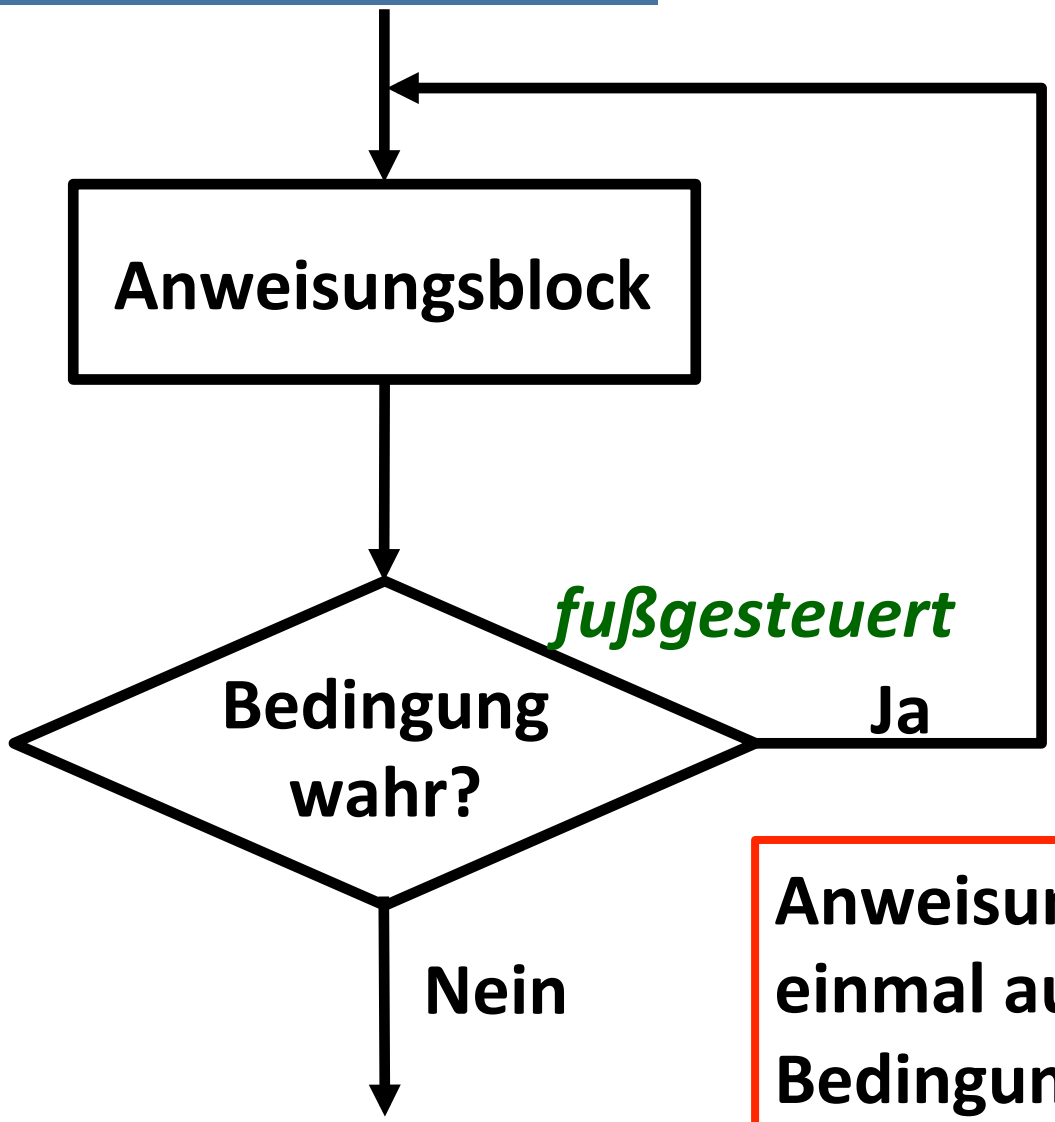
Wie oft wird die folgende Schleife durchlaufen?

```
int u = 10;

while( u = 10 ){

        u--;
}
```

Ausgabe



do … while - Schleife



fußgesteuert

Unterschied zu while?

Anweisungsblock wird immer einmal ausgeführt, bevor die Bedingung geprüft wird.

do … while - Schleife …realisiert im C Quellcode….

```
int n = 5, fakultaet = 1;;

do
{
    fakultaet *= n;
    n--;
}while( n > 1 );

printf("Fakultaet: %d\n", fakultaet);
```

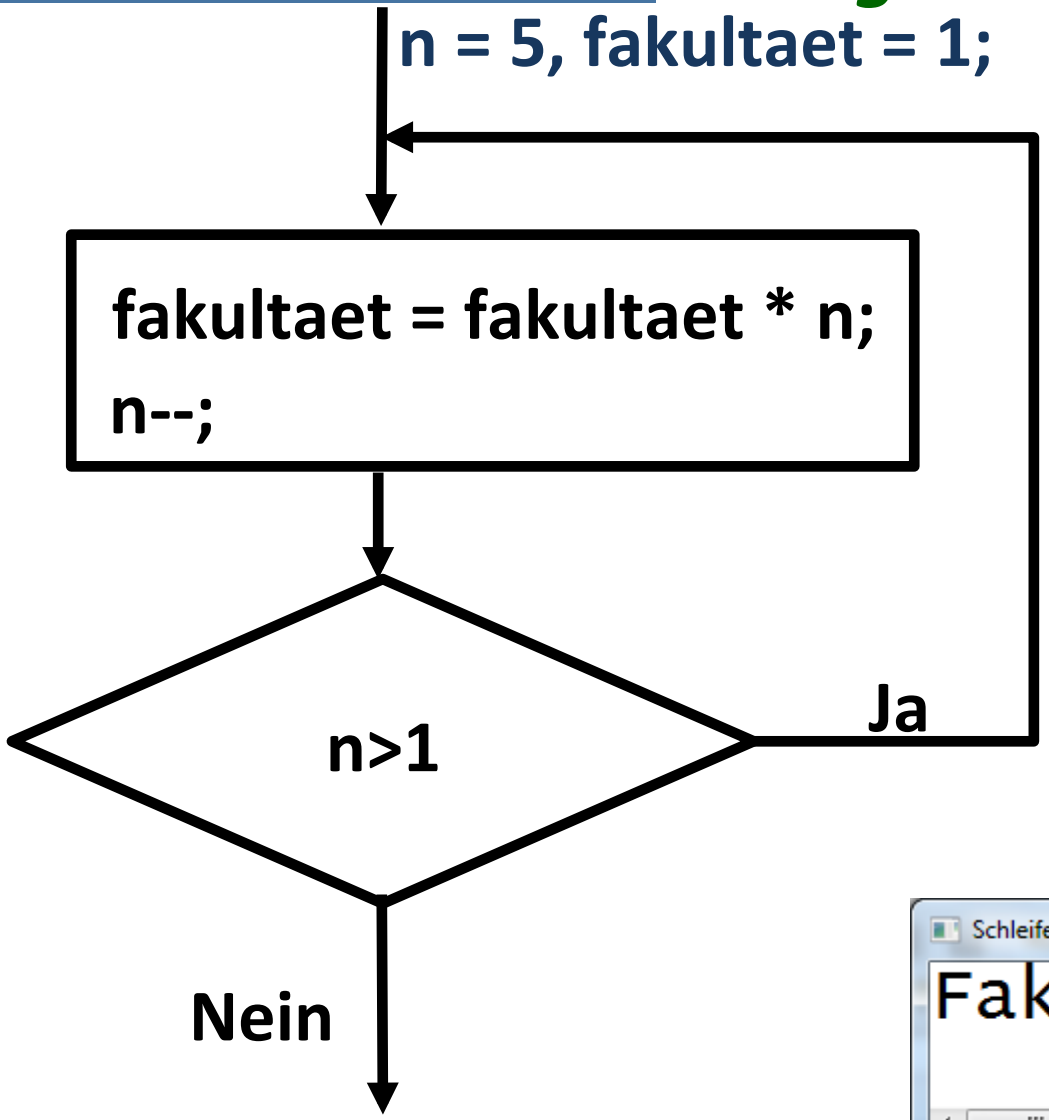
Schlüsselwort

Anweisungsblock

Prüfbedingung für einen erneuten Schleifendurchlauf

Abschluss mit Semikolon

do … while - Schleife …Ausgabe….



Werte jeweils nach:

1. Durchlauf:

n = 4, fakultaet = 5;

2. Durchlauf:

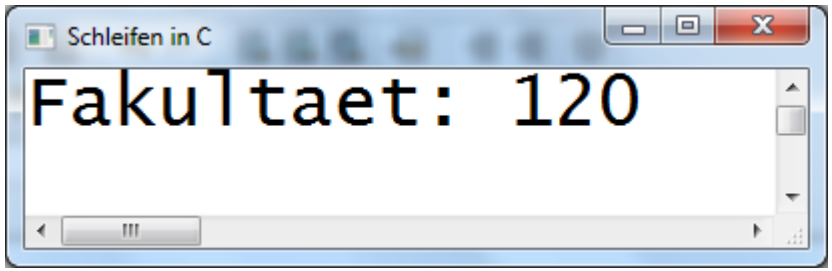
n = 3, fakultaet = 20;

3. Durchlauf:

n = 2, fakultaet = 60;

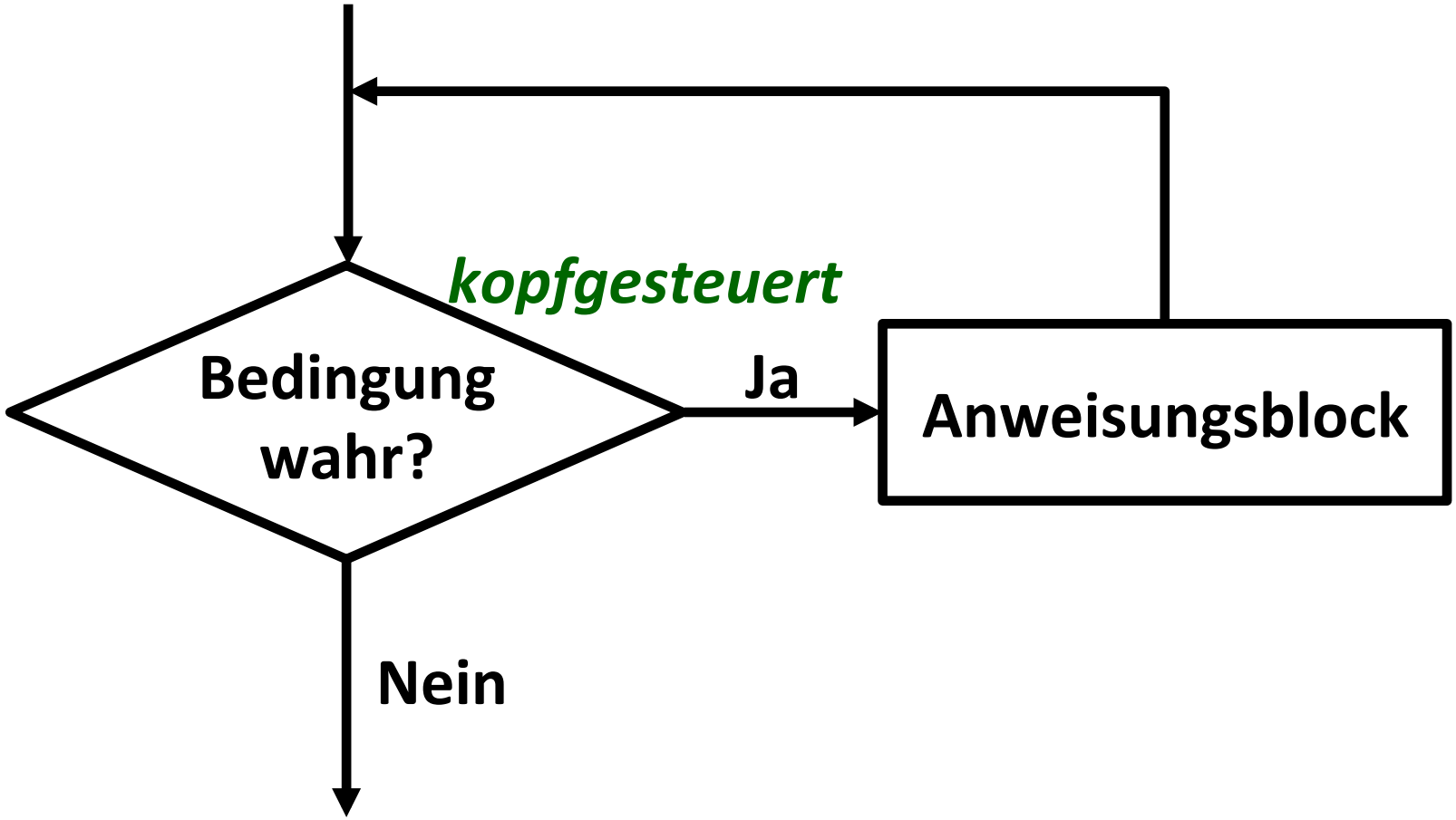
4. Durchlauf:

n = 1, fakultaet = 120;



for - Schleife

Ebenfalls kopfgesteuert

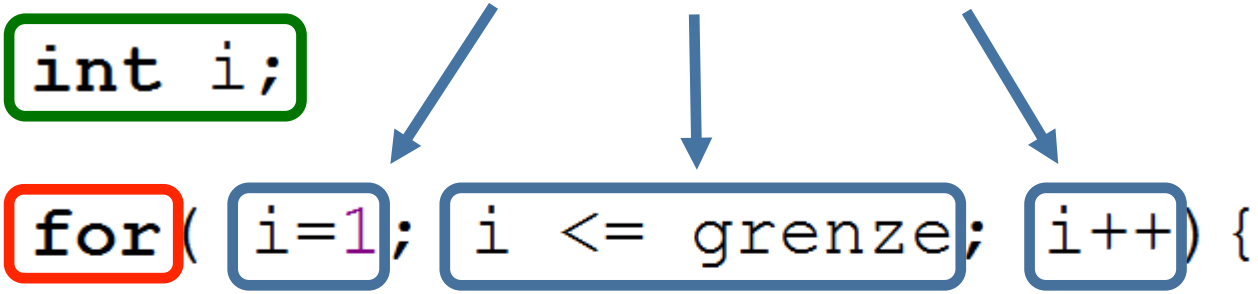


Anweisungsblock wird ausgeführt, so lange die Prüfbedingung true liefert.

for - Schleife *...realisiert im C Quellcode....*

vorher deklarieren

Steuerung durch 3 Komponenten



```
int i;

for( i=1; i <= grenze; i++) {
        summe = summe + i*i;
}

printf("Summe: %d\n", summe);
```

for - Schleife *...Komponente 1: Initialisierung.*

Deklaration und / oder Initialisierung der Laufzeitvariablen

```
int i;
```

Laufvariable vorher deklarieren

```
for( i=1; i <= grenze; i++){
```

Initialisierung

for - Schleife *...Komponente 2: Fortsetzung...*

Prüfbedingung, ob der Anweisungsblock der for-Schleife ein weiteres Mal durchlaufen wird.

```
int i;

for( i=1; i <= grenze; i++){
```

Fortsetzungsbedingung

for - Schleife *...Komponente 3: Aktualisierung...*

Aktualisieren der Laufzeitvariablen für den nächsten Durchlauf.

```
int i;

for( i=1; i <= grenze; i++) {
```

Aktualisierung

for - Schleife *...Syntax...*

Trennung durch Semikolon der Bereiche:

Initialisierung,

Laufzeitbedingung

Aktualisierung

```
for( i=1; i <= grenze; i++){
```

for - Schleife ...Beispiele...

Innere und äußere Schleife

```
int i=0,j=0;

for( i=1; i <11; i++){
```

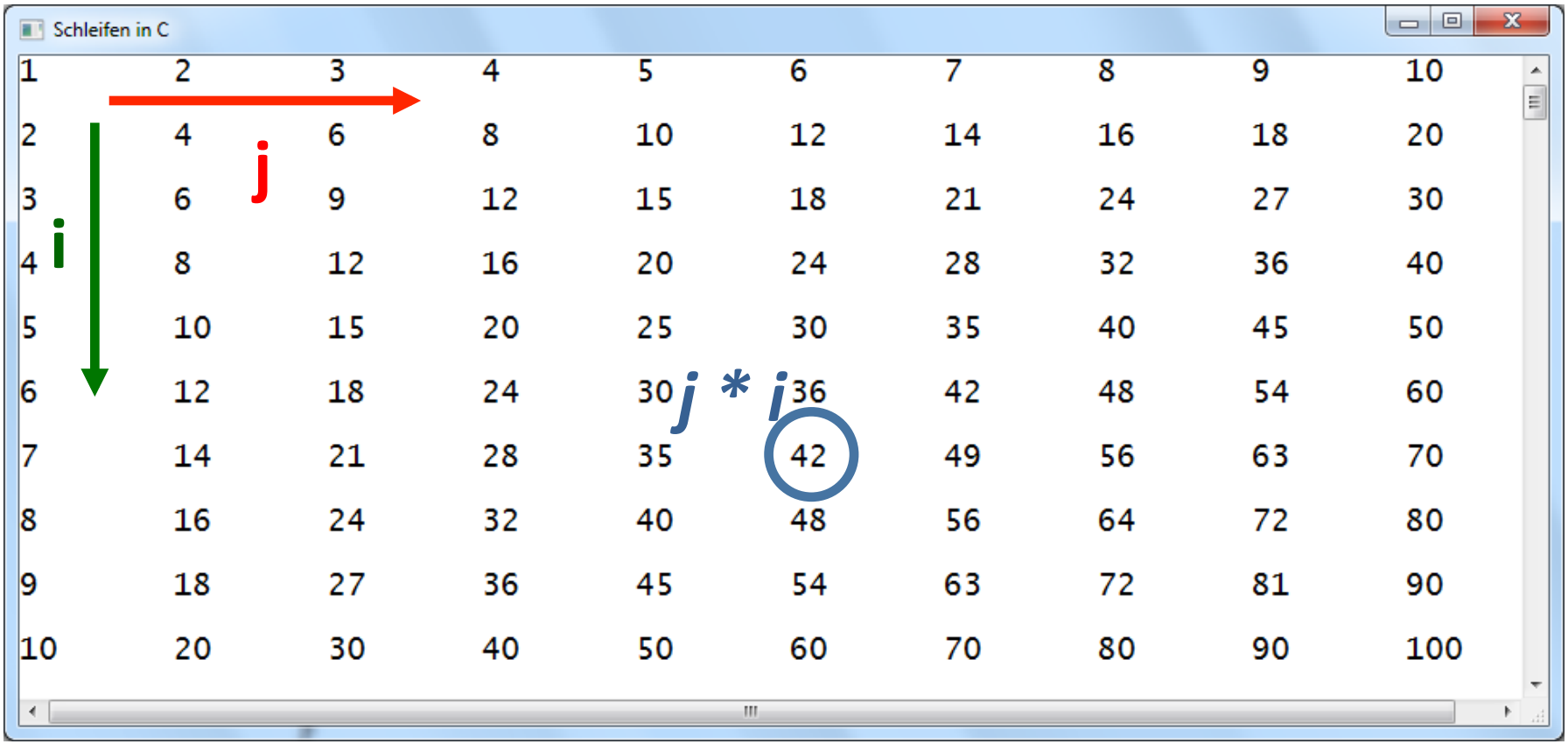
innere Schleife

```
    for( j=1; j <11; j++){

        printf("%d\t",i*j);
    }

    printf("\n");
}
```

Tabulator-Sprung

for - Schleife ...Beispiele...

Ausgabe



for - Schleife ...Beispiele...

```
for( min = max = n; n > 0; ){

      if( n < min){

            min = n;

      }else{

            if( n > max ){

                  max = n;
            }
      }

      printf("Eingabe: ");
      scanf("%d", &n);
}
```

Aktualisierung leer

Initialisierung bereits deklarierter Variablen

Eingabe steuert Laufzeitbedingung

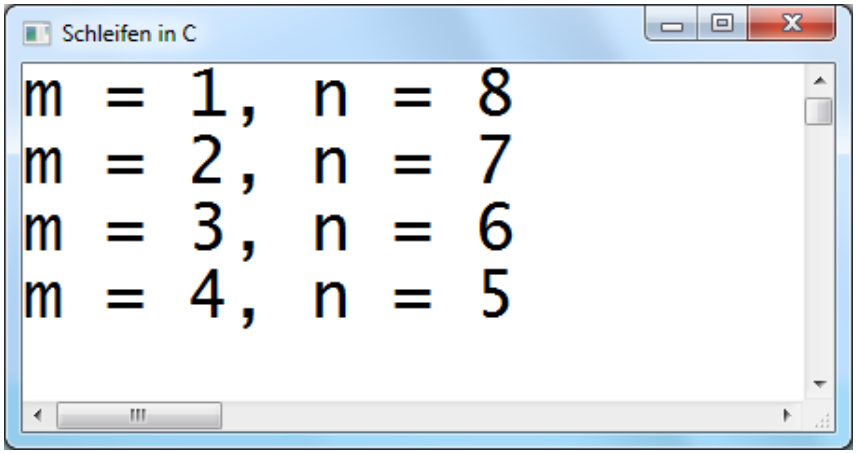
for - Schleife ...Beispiele...

Initialisierung mehrerer Variablen

Aktualisierung von m und n

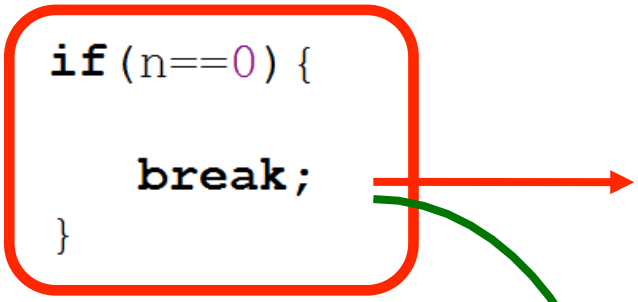
```
for( m=1, n=8; m < n; m++, n-- ){
    printf("m = %d, n = %d\n", m, n);
}
```

Ausgabe?



break ...Der Ausstieg aus einer Schleife...

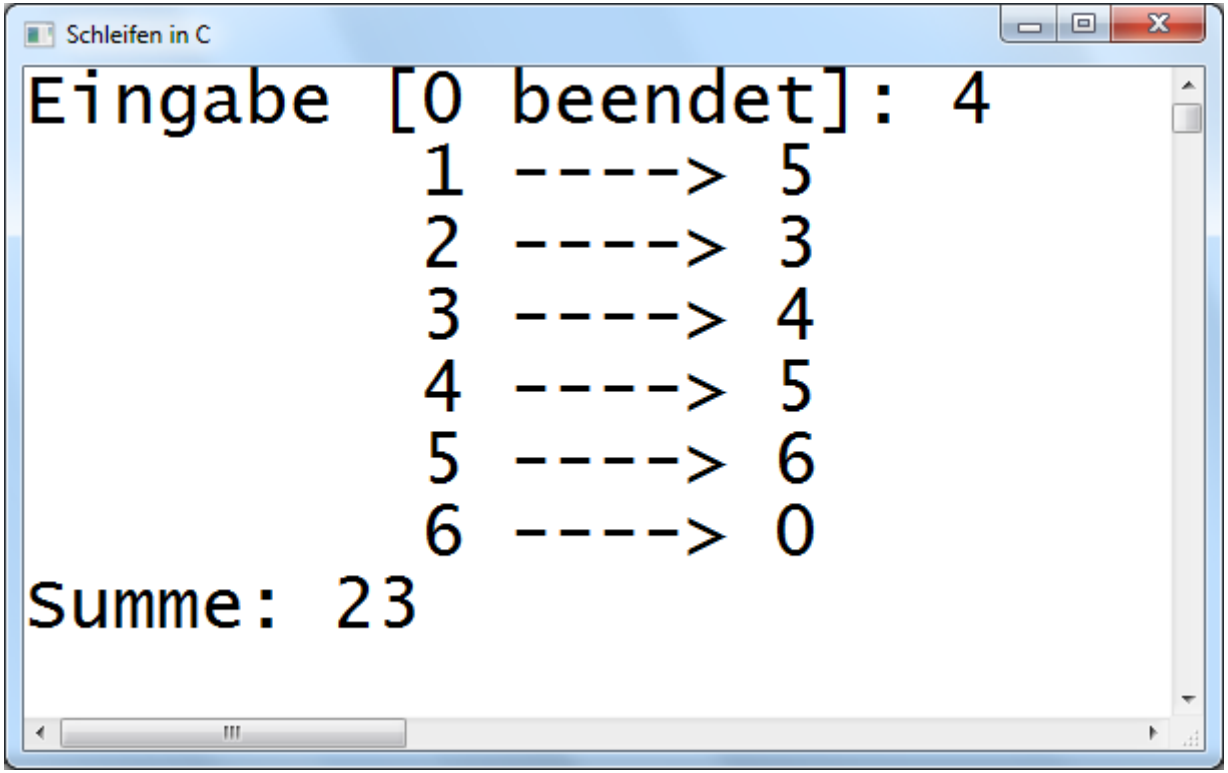
```
int n, count = 0, sum = 0;

printf("Eingabe [0 beendet]: ");
scanf("%f",&n);

for( ; ; ){

      printf("\t %d ----> ", count+1);
      scanf("%d",&n);

      if(n==0){

         break;
      }

      ++count;
      sum+=n;
}

printf("Summe: %d", sum);
```



Sofortiges Verlassen des Anweisungsblocks der Schleife

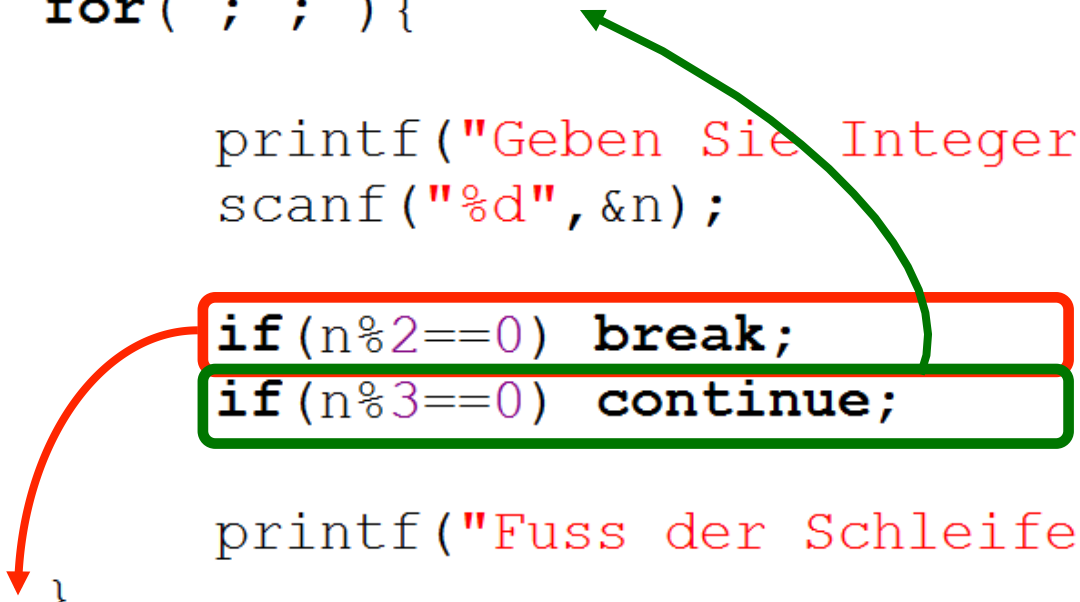
break *...Der Ausstieg aus einer Schleife...*

Ausgabe



continue *...Der Sprung an den Schleifenanfang...*

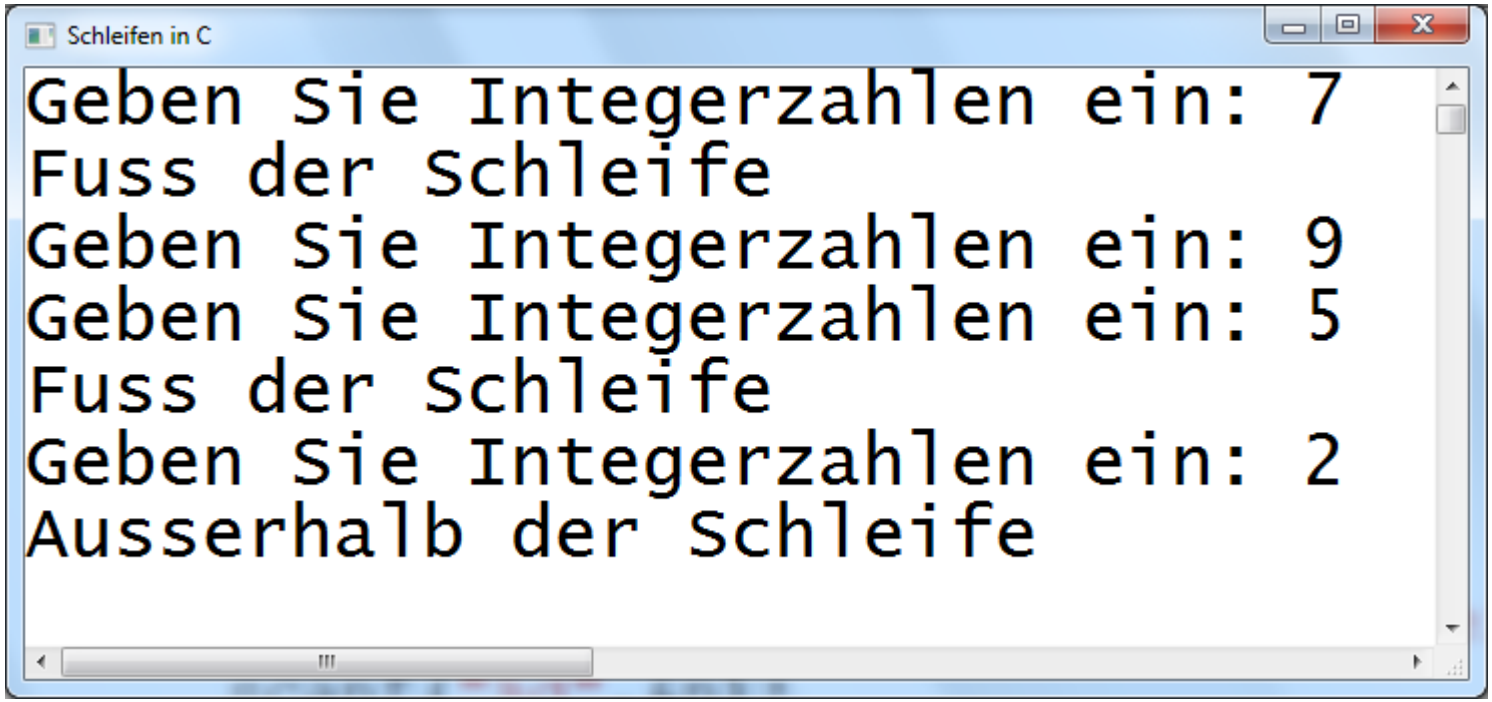
```
int n=0;

for( ; ; ){

      printf("Geben Sie Integerzahlen ein: ");
      scanf("%d",&n);

      if(n%2==0) break;
      if(n%3==0) continue;

      printf("Fuss der Schleife\n");
}

printf("Ausserhalb der Schleife\n");
```



Der Befehl continue veranlasst einen Sprung zum Beginn des Anweisungsblockes der Schleife.

continue …Der Sprung an den Schleifenanfang…

Ausgabe

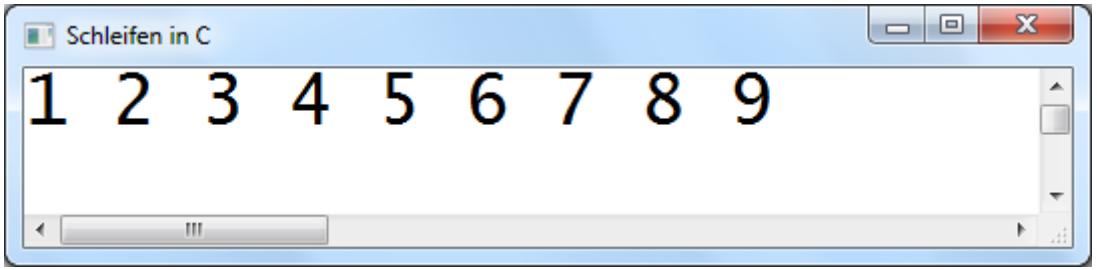



```
if(n%2==0) break;
if(n%3==0) continue;
```

goto …Der Sprung zu einer definierten Marke…

```
int i=0, j=0;

for( ; i<10 ; i++ ){

      for( ; j<10 ; j++ ){

            if( (i+j)%2==0 ) goto esc;
            printf("%d ", i);
      }

esc:;
}
```

Ausdruck selbst definiert!



goto …von Sprung zu Sprung…

```
int n = 0;

printf("Eingabe: ");
scanf("%d", &n);

s1: printf("Schritt 1: n = %d\n", n);
    n--;
    if( n<2 ) goto s5;

s2: printf("Schritt 2: n = %d\n", n);
    if( n<7 ) goto s4;

s3: printf("Schritt 3: n = %d\n", n);
    if( n%2==0 ) goto s1;

s4: printf("Schritt 4: n = %d\n", n);
    n-=2;
    if( n<4==0 ) goto s1;
    else goto s3;

s5:;
```