

Boolsche Ausdrücke und Schleifen

```
/* a) */
(2 <= 3)
/* b) */
(3 * 4 < 12)
/* c) */
int a = 10, b = 10;
(a >= b && b <= a)
/* d) */
int a = 10, b = 9;
(a < b || b < a)
/* e) */
int a = 10, b = 9;
((a == b) || (b == a)) || (a == b++)
/* f) */
int a = 10, b = 9;
((a == b) || (b == a)) || (a == ++b)
```

- 1) Geben Sie für die Ausdrücke a) bis f) jeweils an, ob sie auf true oder false abgebildet werden.

```
/* a) */
for (int i = 0; i < 5; i++) {
    /* .. */
}
/* b) */
for (int i = 0; i < 5; i += 2) {
    /* .. */
}
/* c) */
int i = 2;
do {
    /* .. */
} while (i < 2);
```

- 2) Geben Sie für die Ausdrücke a) bis c) jeweils an, wie oft der Schleifenrumpf ausgeführt wird.

Rekursive Funktionen

Gegeben seien folgende C-Funktionen.

```
unsigned long long f(unsigned n) {  
    if (n == 0)  
        return 1;  
    else  
        return n * f(n - 1);  
}  
  
unsigned g(unsigned a, unsigned b) {  
    if (a == 0)  
        return b;  
    else  
        return g(a - 1, b + 1);  
}  
  
unsigned h(unsigned n) {  
    if (n == 0)  
        return 0;  
    else  
        return n + h(n - 1);  
}
```

a) Geben Sie folgende Werte an:

- i) $f(5)$
- ii) $g(4, 7)$
- iii) $h(10)$

b) Geben Sie für **f**, **g** und **h** jeweils eine mathematische äquivalente Funktion an

c) Schreiben Sie die Funktion **h** mit Hilfe eines for-Loops statt Rekursion um den Stack zu entlasten.