

2.7.A Funktionen/Rekursive Aufrufe – Arbeitsblatt

Aufgabe 1: Einfache Funktionen mit Rekursion

Hier sind ein paar einfache Fälle, die dazu helfen, ein Gespür für Programmierung mit Rekursion zu bekommen. Bei diesen einfachen Fällen bringt Rekursion meist keinen Vorteil, sie dienen eben nur der Übung.

Ein Beispiel ist die Funktion `plus()`: Normalerweise könnte man sie so programmieren:

```

1 unsigned int plus(unsigned int a, unsigned int b)
2 {
3     unsigned int erg;
4     erg = a + b;
5     return erg;
6 }
```

Hätte man eine CPU, die nur plus eins rechnen kann, sähe die Lösung so aus:

```

1 unsigned int plus(unsigned int a, unsigned int b)
2 {
3     while(b>0)
4     {
5         ++a;
6         --b;
7     }
8     return a;
9 }
```

Will man jetzt die Schleife entfernen, kommt man auf die rekursive Lösung:

```

1 unsigned int plus(unsigned int a, unsigned int b)
2 {
3     if(b>0)
4     {
5         unsigned int erg;
6         erg = plus(a, b-1) + 1;
7         return erg;
8     }
9     else
10    return a;
11 }
```

Die Lösung basiert darauf, dass z. B. $a + 4$ um 1 größer ist als $a + 3$, das man wiederum aus $a + 2$ berechnen kann usw. Sie sollen jetzt auf ähnliche Weise einige weitere Funktionen erstellen, die gleichfalls rekursiv arbeiten:

- a) `minusrekursiv.c`: Erstellen und testen Sie die Funktion `minus()`!
Die Formel lautet $a - b = a - 1 - 1 - \dots - 1$. Eins wird b-mal von a abgezogen.

```

1 int minus(int a, unsigned int b);
```

- b) `malrekursiv.c`: Erstellen und testen Sie die Funktion `mal()`!
Die Formel lautet $a \cdot b = a + a + \dots + a$. a wird b-mal aufaddiert.

```

1 int mal(int a, unsigned int b);
```

- c) `durchrekursiv.c`: Erstellen und testen Sie die Funktion `durch()`!
Die Formel lautet $a - b - \dots - b = 0$. b wird x -mal von a abgezogen, bis eine Zahl zwischen $a-1$ und null erreicht wird. x ist dann das Ergebnis.

```
1 int durch(int a, unsigned int b);
```

- d) `hochrekursiv.c`: Erstellen und testen Sie die Funktion `hoch()`!
Die Formel lautet $a^b = a \cdot a \cdot \dots \cdot a$. Eins wird b -mal mit a multipliziert.

```
1 int hoch(int a, unsigned int b);
```