

Übungen 9 Funktionen

1. Aufgabe

a) Schreiben Sie eine Funktion `summe()` mit vier Parametern, die die Summe der Argumente berechnet und zurückgibt.

Parameter: Vier Variablen vom Typ `long`

Return-Wert: Die Summe vom Typ `long`

Deklarieren Sie die zwei letzten Parameter der Funktion `summe()` mit dem DefaultArgument 0. Testen Sie die Funktion `summe()` mit allen drei Aufrufmöglichkeiten. Verwenden Sie dabei als Argument gabzzahlige Zufallszahlen.

b) Strukturieren Sie Ihr Programm nun so, dass die Funktion `main()` und `summe()` jeweils in einer eigenen Quelldatei enthalten sind. z.B. in `main.cpp` und `summe.cpp`

2. Aufgabe

a) Schreiben Sie eine inline-Funktion `Max(double x, double y)`, die das Maximum von `x` und `y` zurückgibt (schreiben Sie `Max` statt `max`, um Konflikte mit anderen Definitionen von `max` zu vermeiden). Testen Sie die Funktion mit Werten, die Sie von der Tastatur einlesen.

Kann die Funktion `Max()` auch mit zwei Argumenten vom Typ `char`, `int` oder `long` aufgerufen werden?

b) Überladen Sie nun `Max()` durch eine weitere inline-Funktion `Max(char x, char y)` für Argumente vom Typ `char`.

Kann die Funktion `Max()` noch mit zwei Argumenten vom Typ `int` aufgerufen werden?

3. Aufgabe

Die Fakultät $n!$ einer positiven ganzen Zahl n ist definiert durch:

$$n! = 1 * 2 * 3 \dots * (n - 1) * n$$

Ferner ist: $0! = 1$

Schreiben Sie eine Funktion, die die Fakultät einer Zahl berechnet.

Argument: Eine Zahl n vom Typ `unsigned int`

Return-Wert: Die Fakultät $n!$ Vom Typ `long double`

Formulieren Sie zwei Versionen der Funktion, wobei die Fakultät

a) mit Hilfe einer Schleife

b) rekursiv berechnet wird

Testen Sie beide Funktionen, indem Sie die Fakultät der Zahlen 0 bis 20 auf dem Bildschirm ausgeben