

---

Programmierübung zur Vorlesung  
**Objektorientierte Programmierung mit C++**  
Sommersemester 2021

**Aufgabenblatt 4: Reihungen und Zeichenketten**

---

Ausgabe: 17.05.2021

Abgabe: 25.05.2021

---

**Aufgabe 1 Vektor-Funktionen**

(5 Punkte)

Felder vom Typ `vector<double>` mit beliebig vielen Werten lassen sich natürlich auch als Vektoren im Sinne der Mathematik betrachten.

Schreiben Sie folgende Vektor-Funktionen:

```
double max(const vector<double> &v)
```

Die Funktion gibt den größten im Vektor auftretenden Wert zurück.

```
bool allPositive(const vector<double> &v)
```

Die Funktion ermittelt, ob alle Werte des Vektors positiv sind.

```
double product(const vector<double> &v1, const vector<double> &v2)
```

Die Funktion gibt das Skalarprodukt zweier Vektoren zurück.

```
vector<double> product(const vector<double> &v, double f)
```

Die Funktion gibt das Produkt des Vektors mit der Zahl zurück.

```
double norm(const vector<double> &v)
```

Die Funktion gibt die Norm eines Vektors zurück ( $\|x\|_2 = \sqrt{\sum_i x_i^2}$ ).

```
void normalize(vector<double> &v)
```

Die Funktion normalisiert einen übergebenen Vektor.

Testen Sie Ihre Funktionen in einem passenden Hauptprogramm. Lassen Sie dazu den Nutzer beliebig große Vektoren eingeben.

**Aufgabe 2 Konvertierung einer Zeichenkette in eine Zahl**

(5 Punkte)

Eine Funktion soll eine übergebene Zeichenkette des Typs `string` in eine ganze Zahl vom Typ `int` konvertieren. Dabei soll sie folgende Schnittstelle aufweisen:

```
int convert(const string &s);
```

Darüber hinaus sollen diese Anforderungen an die Funktion gelten:

- Ein korrekter Eingabestring enthalte eine ganze Zahl ohne Vorzeichen in dezimaler Darstellung.
- Leerzeichen vor der Zahl werden ignoriert.
- Nach der Zahl dürfen Leerzeichen und danach beliebige weitere Zeichen folgen.
- Bei Übergabe eines korrekten Eingabestrings soll die Funktion den Wert der Dezimalzahl zurückgeben.
- Ist der Eingabestring nicht korrekt, so ist eine Fehlermeldung auszugeben und der Wert  $-1$  zurückzugeben.

Beispiele:

Eingabestring	Rückgabewert	Ausgabe
"0"	0	
"128"	128	
" +128"	-1	Fehler
"12x28"	-1	Fehler
"128+x"	-1	Fehler
"128 Punkte"	128	

*Hinweise:*

- Die Integerzahl lässt sich aufbauen, indem einer *leeren* Zahl 0 die weiteren gefundenen Ziffern angehängt werden (siehe Hornerschema).

Beispiel - Konvertierung von `s="128"`:

	0
<code>s[0]='1'</code>	1
<code>s[1]='2'</code>	12
<code>s[2]='8'</code>	128

- Sie sollten überprüfen, ob ein Zeichen `c` eine Ziffer darstellt.
- Benutzen Sie zur Eingabe der Zeichenkette die Funktion `getline(std::cin, s)`, damit auch Leerzeichen eingegeben werden können.