

### Aufgabe 1: Minimax

- Schreiben Sie ein Programm, das mit einer for-Schleife zehn beliebige Zufallszahlen in einem int-Vector speichert und die Werte danach mit einer foreach-Schleife ausgibt.
- Schränken Sie die Zufallszahlen auf das Intervall [-1000; 1000] ein.
- Ermitteln Sie die Summe aller Zahlen im Vector sowie die größte und die kleinste Zahl.
- Verlagern Sie die Berechnung der Summe in eine Funktion mit dem folgenden Prototyp:  

```
int summeVector(const vector<int>& v)
```

### Aufgabe 2: Lottozahlen

- Benutzen Sie wieder eine for-Schleife, um in einen Vector sechs Zufallszahlen aus dem Intervall [1; 49] zu schreiben.
- Fügen Sie eine foreach-Schleife hinzu, die den Inhalt des Vectors auf dem Bildschirm ausgibt.
- Ergänzen Sie den Code so, dass keine Zahl doppelt vorkommen kann. Schließlich gibt es keine zwei gleichen Zahlen in einer Lottoziehung. Zum Testen können Sie den Wertebereich der Zufallszahlen verkleinern.

### Aufgabe 3: Umsortieren

Viele Programme ermöglichen es, die Reihenfolge einer Auflistung auf dem Bildschirm umzukehren. Das können wir auch - wenn auch vorläufig noch auf der Textkonsole.

Der Benutzer soll beliebig viele Zahlen vom Typ double in einen Vector eingeben. Die Eingabe wird durch den Wert 0 beendet.

- Erst die einfache Variante: Geben Sie die Zahlen im Vector von hinten nach vorne aus.
- In der Praxis benötigt man oft tatsächlich die umgekehrte Reihenfolge im Arbeitsspeicher. Benutzen Sie einen zweiten Vector als Zwischenspeicher, um die Zahlen rückwärts wieder im ersten Vector abzulegen.
- Schreiben Sie eine zweite Variante des Programms, das zum Umsortieren einen Stack als Zwischenspeicher verwendet.

Welche Version finden sie leichter verständlich? Ist eine der beiden schneller?

### Zusatzaufgabe zum Üben

Elektrische Widerstände kann man als fertige Bauteile kaufen, aber nicht mit beliebigen Werten, sondern nur in sogenannten Normreihen.

- Legen Sie zunächst einen Vector für die Widerstands-Normreihe E6 an. Sie enthält die Werte 100, 150, 220, 330, 470, 680 und 1000. Lassen Sie dann eine Zahl zwischen 100 und 1000 einlesen und den nächstliegenden Normwert nach E6 suchen und ausgeben.
- Verlagern Sie die Suche des Normwerts in eine Funktion, der Sie den Vector und den vom Benutzer eingegebenen Widerstandswert übergeben. Die Ein- und Ausgabe bleibt im Hauptprogramm.
- Von den elektrischen Widerständen gibt es nicht nur die Werte aus der Normreihe, sondern jeweils auch ein Hundertstel, ein Zehntel, das Zehnfache, das Hundertfache und so weiter.

Erweitern Sie das Programm auf beliebige Eingaben:

- Eingabe von beliebigen positiven Zahlen vom Typ double.
- Umrechnung der Zahl durch mehrfache Multiplikation oder Division mit 10 in den Bereich 100...1000.
- Normwert bestimmen und wieder auf die richtige Zehnerpotenz umrechnen.

Beispiel:

Eingabe 51420.2 => /10/10 ergibt 514.202 => Normwert 470 => \*10\*10 => Ausgabe 47000

### Expertenaufgabe

- Untersuchen Sie für verschiedene Intervalle, wie die Zahlen, die der Zufallszahlengenerator mit der Modulo-Division liefert, verteilt sind; das heißt, wie oft jede Zahl vorkommt.
- Überlegen Sie sich eine bessere Methode, um mit dem von uns verwendeten Generator verschiedene Intervalle zu erhalten.