Listen

1 Listen

1. Implementieren Sie eine Funktion `add(point_a, point_b)`, die zwei Punkte eines karthesischen Koordinatensystems addiert. Es gilt `[a, b] + [x, y] = [a + x, b + y]`.
2. Schreiben Sie eine Funktion `dist(point_a, point_b)`, die die Distanz zwischen zwei Punkten in einem karthesischen Koordinatensystems zurückgibt.
3. Bestimmen Sie das Minimum einer Liste numerischer Werte (Integer und Float).
4. Füllen Sie eine Liste mit einem Satz Pokerkarten. Speichern Sie Karten als Listen der Form `[value, suit]`. Welche Datentypen verwenden Sie für `value` und `suit`?
5. Schreiben Sie eine Funktion `sub(list_a, list_b)`, die `list_b` von `list_a` „subtrahiert“. Mit anderen Worten: Es sollen all jene Elemente von `list_a` zurückgegeben werden, die nicht in `list_b` enthalten sind.

2 Zusatzübungen

Sollten Sie bereits mit allen Übungen fertig sein, können Sie versuchen, folgende Zusatzprobleme zu lösen:

1. Füllen Sie eine Liste der Länge 20 mit Zufallszahlen. Überlegen Sie sich einen einfachen Algorithmus, mit dem eine Liste nur mit Vergleich- und Vertauschoperationen sortiert werden kann. Wie könnte ein solcher Algorithmus aussehen? Versuchen Sie zuerst einen Algorithmus zu entwickeln, bevor Sie etwas implementieren! Sie sollten insbesondere in der Lage sein, einem Bekannten zu erklären, wie dieser Algorithmus funktioniert!

   Erst wenn Sie schon eine konkrete Vorstellung ihres Algorithmus haben, implementieren Sie ihn. Testen Sie ihre Implementierung mit der zufällig gefüllten Liste! Verwenden Sie *keine* vorimplementierte Sortierfunktion. *Tipp:* Sie können Zufallszahlen mit der Funktion `random` aus dem Modul `random` erzeugen (`http://docs.python.org/3/library/random.html`).
2. Implementieren Sie eine Funktion, mit der Sie das Kartenspiel aus der obigen Aufgabe „mischen“ können.
3. Implementieren Sie eine binäre Uhr, die die aktuelle Uhrzeit ausgibt (siehe auch `https://de.wikipedia.org/wiki/Bin%C3%A4re_Uhr`). Suchen Sie zuerst einen passenden Befehl, der Ihnen die aktuelle Systemzeit zurückgibt. Wandeln Sie dann sowohl Stunde (0–23) als auch Minute (0–59) in eine Binärzahl um. Üblicherweise verwendet man binäre LEDs, die leuchten oder eben nicht. Bilden Sie dieses Verhalten nach, indem Sie ein X für leuchtend, und ein O für nichtleuchtend auf der Kommandozeile ausgeben. Beispiel:

```
 OXOXX (8 + 2 + 1 = 11 Uhr)
OXOXXX (16 + 4 + 2 + 1 = 23 Minuten)
```