Ü B U N G E N zu Programmiersprache 1 (Python, C++) WS 2021/2022

Übungsblatt Nr. 7: in der Übungsstunde

1) a) Lesen Sie in einer Endlos-Schleife Integerpaare a, b ein und geben Sie diese aus. Die Schleife soll enden, wenn nichts eingegeben wurde.
   b) Erzeugen Sie aus den eingelesenen Zahlenpaaren den String `filename = "from_a_to_b.txt"` und geben Sie ihn aus.
   c) Erzeugen Sie die Liste `X = [0.000, 0.001, 0.002, 0.003, ... , 10.000]`. `Y` sei die Liste aller Funktionswerte von `X` unter $f(x) = x\sin(x)$. Schreiben Sie in die Datei mit Namen `filename` alle Wertepaare x_i, y_i mit $a \leq y_i \leq b$. Verwenden Sie Format `6.3f` für `x` und `12.8f` für `y`.

2) a) C++: Lesen Sie in einer `while`-Schleife Integer ein und geben Sie diese wieder aus. Die Schleife soll enden, wenn ein Buchstabe eingelesen wurde oder wenn das Inputende erkannt wurde (STRG-Z unter Windows, STRG-D unter Unix).
   b) Nehmen Sie statt des `while` eine `for`-Schleife.
   c) Geben Sie die Zahlen 1000, 999, 997, 994, 990, ... der Reihe nach aus. Der Abstand der Zahlen erhöht sich bei jeder Zahl um 1. Stoppen Sie, bevor eine Zahl < -1000 ausgegeben wird.
   d) Erzeugen Sie einen `std::vector<int> v` aus den Zahlen der Aufgabe c). Geben Sie diesen Container mit einem `range-for` aus.

Übungsblatt Nr. 7 Hausübung: Die folgenden Aufgaben sind Pflicht und zählen 1 Punkt!

1) Python:
   a) Lesen Sie den Datenfile `"data_99_3.dat"` als Liste von Zeilen in ihr Programm ein. Speichern Sie alle Zeilen in der Datei `"sorted.dat"`, in der die 3 Zahlen der Zeile aufsteigend sortiert waren. Verwenden Sie ein Prädikat `is_sorted(line)`, die eine Zeile in die 3 Zahlen zerlegt und diese Bedingung testet.
   b) Schreiben Sie analog alle Zeilen in die Datei `"even.dat"`, die nur aus geraden Zahlen besteht. Die Zeilen sollten in umgekehrter Reihenfolge zur Inputdatei sein.

2) C++: Verwenden Sie `while`-Schleifen
   a) um die Zahlen 2, 4, 6, 8, ... , 50 auszugeben.
   b) um die Buchstaben z, y, x, w, ... , a auszugeben.

3) C++: Verwenden Sie `for`-Schleifen, um die vorige Aufgabe zu lösen.

4) Nehmen Sie den Container `std::vector<int> v` aus dem Demobeispiel der Übung. Kombinieren Sie `range-for` und `if` und geben Sie alle Zahlen des Containers aus
   a) die gerade sind.
   b) die im Intervall $[-50, 50]$ liegen.
   c) die Vielfache von 7 oder 11 sind.

Extra-Aufgabe(n): Diese Aufgaben sind freiwillig und zählen 2 Punkte!

5) Verwenden Sie wieder den Container `std::vector<int> v` aus dem Demobeispiel. Erzeugen Sie die Container `v6, v3, rest`, wobei `v6` alle Vielfachen von 6, `v3` die Vielfachen von 3 die nicht durch 6 teilbar sind und `rest` alle übrigen enthält. Geben Sie alle 3 Container zur Kontrolle aus.

6) Lösen Sie die Hausaufgabe Nr. 2 der 3. Übungsstunde (Funktion `sum35(n)`) mit C++.