Dr. Andreas Borchert
Norbert Heidenbluth
Abteilung Angewandte Informationsverarbeitung

02.02.2005
Blatt 14

Allgemeine Informatik I für Mathematiker/Wirtschaftsmathematiker (WS 2004/2005)

Abgabetermin: 09. Februar 2005

Aufgabe 24: Shellsort für Auntie Anne (6+4+5 Punkte)

Erinnern Sie sich noch an Auntie Anne aus dem zweiten Übungsblatt? Ja? Gut, dann haben Sie Auntie Anne etwas voraus, denn die alte Dame ist leider etwas vergeßlich und weiß partout nichts mehr von dem Gefallen, den Sie ihr im letzten November erwiesen haben.

Unter anderem haben Sie damals für Auntie Annie eine Statistik ihrer beliebtesten Urlaubsziele sortiert darstellen lassen. Genau das möchte Ihre Erbtante von Ihnen nun schon wieder.

Damals haben Sie diesen Wunsch durch das Kommando “`sort`” auf der Shell erfüllen können, diesmal erledigen Sie das Sortieren durch die Implementierung des sogenannten *Shellsort* (Bitte beachten Sie dieses grandiose Wortspiel ;-))

Die *Datei mit den Urlaubszielen* verwendet das folgende Format:

```
#xZiel
```

wobei `#` für die Häufigkeit (als Integer-Zahl) steht, mit der das Ziel besucht worden ist, also zum Beispiel:

```
10xRom
5xParis
...
```

Bei Auntie Anne finden Sie im Regal den dritten Band des Werkes “The Art of Computer Programming” von D. Knuth. Auf Seite 83 ff. wird dort der folgende Algorithmus für den *Shellsort* vorgestellt:

> **Algorithm D** (*Shellsort*). Records $R_1, \ldots, R_N$ are rearranged in place; after sorting is complete, their keys will be in order, $K_1 \leq \cdots \leq K_N$. An auxiliary sequence of increments, $h_{t-1}, h_{t-2}, \ldots, h_0$ is used to control the sorting process, where $h_0 = 1$; proper choice of these increments can significantly decrease the sorting time.
>
> **D1.** [Loop on s.] Perform step D2 for $s = t-1, t-2, \ldots, 0$; then terminate the algorithm.
>
> **D2.** [Loop on j.] Set $h \leftarrow h_s$, and perform steps D3 through D6 for $h < j \leq N$. (We will use a straight insertion method to sort elements that are h positions apart, so that $K_i \leq K_{i+h}$ for $1 \leq i \leq N-h$.
>
> **D3.** [Set up i, K, R.] Set $i \leftarrow j-h, K \leftarrow K_j, R \leftarrow R_j$.
>
> **D4.** [Compare $K : K_i$.] If $K \geq K_i$, go to step D6.
>
> **D5.** [Move R_i, decrease i.] Set $R_{i+h} \leftarrow R_i$, then $i \leftarrow i-h$. If $i > 0$, go back to step D4.
>
> **D6.** [R into R_{i+h}.] Set $R_{i+h} \leftarrow R$.

Auntie Annie — bekannt für ihre hohen Ansprüche — möchte nun von Ihnen ein Oberon-Programm haben, das aus einer Datei die Urlaubsziele mit den entsprechenden Häufigkeiten einliest und sortiert wieder auf der Standardausgabe ausgibt. Dabei soll sowohl das Sortieren *nach Häufigkeit* als auch *nach Urlaubszielen* möglich sein (steuerbar durch ein entsprechendes Flag beim Aufruf Ihres Programms auf der Kommandozeile).

Als Sie diese Aufgabe hören, brechen Sie weinend zusammen und denken sich: “Schon wieder so ein Riesending!”. Aber da nimmt Sie Onkel Heidi in den Arm und tröstet Sie, indem er Ihnen folgende Gliederung vorschlägt:

Teil a) Implementierung des Sortieralgorithmus (6 Punkte)

Das wichtigste an diesem Übungsblatt ist die Implementierung des Sortieralgorithmus selbst. Schreiben Sie dazu zunächst das entsprechende Oberonprogramm, das unter Verwendung des gegebenen Algorithmus **Zahlen** sortiert.

Teil b) Erweiterung des Programms: Verwendung von Prozedurtypen und Sortieren von Strings (4 Punkte)

Machen Sie sich nun Gedanken, über die Datenstruktur, die Sie für diese Aufgabe benötigen. Verändern Sie Ihren Sortieralgorithmus aus dem vergangenen Schritt so, daß er sowohl Zahlen als auch Strings sortieren kann. An geeigneter Stelle könnten Sie dies zum Beispiel durch die Verwendung von *Prozedurtypen* erreichen und sich dabei am *Beispiel aus dem Skript* orientieren.

Teil c) Einlesen aus Dateien, Argumentverarbeitung (5 Punkte)

Nun fehlt noch das Einlesen der Daten und die Argumentverarbeitung. Alles, was jetzt hierzu noch notwendig ist, können Sie problemlos aus alten Übungsaufgaben und / oder Beispielen übernehmen. Der Zeitaufwand hierfür sollte minimal sein!

Viel Erfolg!