

Übungen zum Programmierkurs I

Abgabe bis zum Freitag, 15.11.2002, 15:00 Uhr

Bitte beachten Sie die allgemeinen Hinweise am Ende des Blattes!

Aufgabe 3.1 **Zahlendarstellung II (mittel)** **5 Punkte**

In Aufgabe 2.3 haben Sie ein Programm geschrieben, welches die Darstellung natürlicher Zahlen zu einer beliebigen Basis b errechnet. Erweitern Sie dieses Programm auf folgende Weise: Es soll die Darstellung einer nicht-negativen reellen Zahl r zu einer Basis b mit t Nachkommastellen berechnet werden, wobei r sowie die natürlichen Zahlen $b > 1$ und t vom Benutzer vorgegeben werden.

Hinweis 1: Geben Sie die Darstellung in der Form `vorkomma . nachkomma` an, wobei `nachkomma` die Länge t hat. Die i -te Stelle von `nachkomma` (von links) habe die Wertigkeit b^{-i} . Da Sie nur eine begrenzte Anzahl von Nachkommastellen ausgeben, ist die ausgegebene Darstellung u.U. nur eine Annäherung von r . Runden Sie die letzte Ziffer daher so auf bzw. ab, dass r möglichst genau dargestellt wird.

Hinweis 2: Zur Darstellung von 'Ziffern' mit einem Wert größer als 10 siehe Aufgabe 2.3.

Aufgabe 3.2 **Teilmengen (mittel)** **5 Punkte**

Bekanntlich gibt es von jeder n -elementigen Menge genau 2^n unterschiedliche Teilmengen, etwa hat $\{1, 2\}$ die $2^2 = 4$ Teilmengen $\emptyset, \{1\}, \{2\}, \{1, 2\}$.

Ihre Aufgabe ist es, einen Algorithmus zu entwerfen und zu programmieren, der zu einem vom Benutzer vorgegebenen Wert von n die 2^n Teilmengen von $\{1, \dots, n\}$ wiederholungsfrei in irgendeiner Reihenfolge aufzählt.

Aufgabe 3.3 **Textsuche (schwer)** **4+6 Punkte**

Schreiben Sie ein Programm, welches zunächst zwei Zeichenketten s und t einliest. Danach soll folgendes geschehen:

- Untersuchen Sie, ob t als Teil der Zeichenkette s vorkommt (z.B. wäre `bb` ein Teil der Zeichenkette `abba`, nicht jedoch `aa`).
- Erweitern Sie das Programm aus Teil (a) so, dass falls t in s vorkommt, alle Vorkommen von t aus s gelöscht werden und die auf diese Weise verkürzte Zeichenkette ausgegeben wird.

Hinweis 1: Es empfiehlt sich, zum Einlesen der Zeichenkette eine andere Methode als in Aufgaben 2.1 und 2.2 zu benutzen. Der GNAT-Compiler bietet die beiden Pakete `Ada.Strings.Unbounded` und `Ada.Strings.Unbounded.Text_IO` an. Diese ermöglichen Ihnen, mit `u:Unbounded_String` eine Zeichenkette unbegrenzter Länge zu deklarieren.

Die Anweisung `u:=Get_Line` liest eine solche Zeichenkette ein, wobei das Ergebnis die Eingabe des Benutzers bis zum ersten Zeilenumbruch enthält. Mit `Length(u)` können Sie die Länge, mit `Element(u,i)` das *i*-te Zeichen in *u* abfragen.

Die erwähnten Pakete enthalten auch eine Funktion namens `Index`, die genau die in (a) verlangte Funktionalität liefert. Diese (und ähnliche Funktionen) sollen Sie selbstredend nicht verwenden, sondern die Funktion selbst ausprogrammieren.

Hinweis 2: Obgleich der Aufgabentext in (b) auf den ersten Blick deutlich genug erscheinen mag, ist die Aufgabe keineswegs in allen Fällen eindeutig spezifiziert. Welche Sonderfälle gibt es? Tip: Überlegen Sie, was etwa bei $s = ababa$, $t = aba$ oder $s = aabb$, $t = ab$ passieren soll. Behandeln Sie Sonderfälle so, wie Sie es für richtig halten, aber dokumentieren Sie das Verhalten Ihres Programms in diesen Fällen.

Allgemeine Hinweise

- Grundsätzlich: Programme sind in Ada95 zu schreiben und müssen fehlerfrei übersetzbar sein!
- Schreiben Sie Ihre Programme so, dass Ihre Wirkungsweise möglichst leicht nachvollzogen werden kann (insbesondere vom Korrektor). Dazu gehören:
 - die Verwendung sinnvoller Variablenamen (oder zumindest die Erklärung, was der Zweck jeder Variablen ist, falls dies aus dem Namen nicht hervorgeht);
 - Kommentare dort, wo der Zweck von Anweisungen nicht unmittelbar offensichtlich ist, insbesondere, wenn mathematische Überlegungen eine Rolle spielen;
 - ein übersichtlicher Programmierstil, beispielsweise sollten die Inhalte von Schleifen oder die Zweige von if-Anweisungen eingerückt werden, damit die Struktur des Programms offenbar wird.
- Die Aufgabenblätter stehen jeweils freitags auf der Webseite und werden Montag in der Vorlesung in Papierform verteilt; sie sind jeweils bis zum nächsten Freitag zu lösen. Die Webseite zum Programmierkurs ist wie folgt:
`http://www.informatik.uni-stuttgart.de/ifi/fk/lehre/ws02-03/ada95/`
- Es wird darauf hingewiesen, dass der Besuch der Übungen grundsätzlich Pflicht ist, da Sie in der Lage sein müssen, Ihre Programme auf Aufforderung in der Übung zu erklären. Sind Sie einmal aus gutem Grunde verhindert, sprechen Sie dies bitte (sofern möglich) zuvor mit Ihrem Tutor ab.
- Pro Aufgabenblatt werden maximal 20 Punkte auf den Übungsschein angerechnet.
- Bei Fragen wenden Sie sich bitte an Ihren Tutor oder an die Übungsleitung:
`Stefan.Schwoon@informatik.uni-stuttgart.de` oder Tel. 7816-427