

☐ Gr. 1, Dr. D. Auer☐ Gr. 2, Dr. G. Kronberger☐ Gr. 3, Dr. H. Gruber

Name _____ Aufwand in h _____

Punkte _____ Kurzzeichen Tutor / Übungsleiter _____ / _____

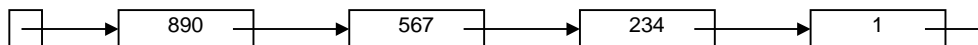
1. Rechnen mit wirklich großen Zahlen zur Basis 1000**(12 Punkte)**

Nachdem Sie in der Übung 4 ja schon mit Zahlen bis zur Basis 36 gerechnet haben, kann Sie nichts mehr erschüttern ;-) Also:

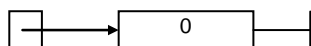
Beliebig große, nicht negative ganze Zahlen (*BigInts*) können realisiert werden, indem man sie (z. B.) zur Basis 1000 darstellt und die einzelnen "Ziffern" (*big digits*, Werte zwischen 0 und 999) in Form einer einfach-verketteten Liste so anordnet, dass die niederwertigste "Ziffer" am Anfang und die höchstwertigste am Ende der Liste steht.

Beispiele:

Die Zahl 1.234.567.890 (Tausenderpunkte hier nur zur Verbesserung der Lesbarkeit) wird durch folgende Liste mit vier Knoten dargestellt:



Die Zahl 0 wird durch die Liste mit einem 0-Knoten dargestellt:



Das Vorzeichen der Zahl kann durch einen Vorzeichenknoten (mit dem Wert +1 oder -1) am Anfang der Liste dargestellt werden. Realisieren Sie mit den unten angegebenen Datentypen *BigInts* und implementieren Sie die Operationen zum Addieren und Multiplizieren.

TYPE

NodePtr = ^Node;

Node = RECORD

next: NodePtr;

val: INTEGER; (*one digit of BigInt: 0 <= val <= 999*)

END; (*RECORD*)

BigIntPtr = NodePtr;

FUNCTION Sum (a, b: BigIntPtr): BigIntPtr; (*compute sum = a + b*)

FUNCTION Product(a, b: BigIntPtr): BigIntPtr; (*compute product = a * b*)

Hinweis: Um Sie von „mühsamer Kleinarbeit“ zu entlasten, stehen für das Einlesen und das Ausgeben von *BigInts* folgende zwei Prozeduren (zusammen mit einigen Hilfsprozeduren und -funktionen) in einem Rahmenprogramm (*BigInts.pas*) fix und fertig zur Verfügung:

PROCEDURE ReadBigInt (VAR bi: BigIntPtr);

PROCEDURE WriteBigInt(bi: BigIntPtr);

2. Christkind auf Rationalisierungsstrip

(2 + 4 + 6 Punkte)

Das Christkind bekommt jedes Jahr viele Wunschzettel von den Kindern, und bisher hat es diese von Hand ausgewertet. Weil auch das Christkind immer älter, die Wünsche aber immer mehr werden, muss es rationalisieren und will wie folgt vorgehen: Die von den Kindern handgeschriebenen Wunschzettel werden eingescannt, mit Schrifterkennungssoftware weiterbearbeitet und alle zusammen in einer Textdatei *Wishes.txt* so abgespeichert, dass in einer Zeile jeweils der Name (*name*) des Kinds (gefolgt von Doppelpunkt und Leerzeichen) und sein Wunsch (*item*) stehen. Die Kindernamen sind eindeutig und dürfen mehrfach vorkommen (für den häufigen Fall, dass ein Kind mehrere Wünsche hat):

Beispiel:

```
...
Christoph: Schlitten
Barbara: Barbie-Puppe
Barbara: Puppenküche
Christoph: Matchboxauto
Barbara: Blockflöte
Susi: Strolchi-Puppe
...
```

Nun wünscht sich das Christkind aber etwas von Ihnen: Ein Pascal-Programm *WLA* (*wish list analyzer*), das nach dem Einlesen der Wünschedatei folgendes ermittelt: eine *Bestellliste* mit den Bestellmengen für die einzelnen Gegenstände, damit es diese schnell und einfach besorgen kann, sowie eine *Zustellliste*, in der für jedes Kind alle Geschenke eingetragen sind, sodass es die Geschenke pünktlich ausliefern kann. Sie beschließen, das Programm *WLA*, mittels einfach-verketteter Listen auf Basis folgender Typdeklarationen zu realisieren:

```
(*for a:*)
WishNodePtr = ^WishNode;
WishNode = RECORD
    next: WishNodePtr;
    name: STRING;
    item: STRING;
END; (*RECORD*)
WishListPtr = WishNodePtr;
```

```
(*for b:*)
OrderNodePtr = ^OrderNode;
OrderNode = RECORD
    next: OrderNodePtr;
    item: STRING;
    n: INTEGER;
END; (*RECORD*)
OrderListPtr = OrderNodePtr;
```

```
(*for c:*)
DelivNodePtr = ^DelivNode;
DelivNode = RECORD
    next: DelivNodePtr;
    name: STRING;
    items: ItemListPtr;
END; (*RECORD*)
DelivListPtr = DelivNodePtr;
```

- Bauen Sie beim Lesen der Wünschedatei die Wunschliste (*wish list*) gemäß obiger Deklarationen (links) so auf, dass die Reihenfolge der Wünsche in der Liste jener in der Datei entspricht.
- Entwickeln Sie eine Funktion *OrderListOf*, die auf Basis einer Wunschliste aus a) eine Bestellliste (*order list*) gemäß obiger Deklaration (Mitte) für das Christkind liefert. In der Bestellliste muss für jeden Gegenstand der Wunschliste (*item*) ein Knoten mit der Häufigkeit (*n*) des Auftretens dieses Gegenstands in der Wunschliste enthalten sein.
- Entwickeln Sie eine weitere Funktion *DeliveryListOf*, die auf Basis einer Wunschliste aus a) eine Zustellliste (*delivery list*) gemäß obiger Deklarationen (rechts) liefert. In der Zustellliste muss für jedes Kind (*name*) ein Knoten mit einer Liste aller Geschenke (*items*) für dieses Kind enthalten sein. Die Datentypen für die Geschenkliste pro Kind (*items*) müssen Sie selbst analog zu allen anderen deklarieren.

Hinweis: Um Ihnen das Abtippen der obigen Deklarationen zu ersparen und ein Rahmenprogramm zur Verfügung zu stellen, welches das Einlesen der Wünschedatei erledigt, haben wir *WLA.pas* und eine einfache Wünschedatei (*Wishes.txt*) in *WLA.zip* für Sie vorbereitet.