W.Nestler

Übung GI (Zahlendarstellung - Teil 3)

1. Die Zahlen **+0**, **-0**, **+∞** , **-∞**, **not a number**, **64.625**, **75.4**, **-2** sind als 32-Bit Gleitpunktzahlen zu schreiben.

2. Die folgenden **32-Bit Gleitpunktzahlen** sind **dual**, **hexadzimal** und angenähert **dezimal** anzugeben:
   - die größte Gleitpunktzahl
   - die kleinste Gleitpunktzahl
   - die kleinste positive Gleitpunktzahl größer als Null
   - die größte negative Gleitpunktzahl

3. Die folgenden **64-Bit Gleitpunktzahlen** sind **dual**, **hexadzimal** und angenähert **dezimal** anzugeben:
   - die größte Gleitpunktzahl
   - die kleinste Gleitpunktzahl
   - die kleinste positive Gleitpunktzahl größer als Null
   - die größte negative Gleitpunktzahl

4. Die Umwandlung einer ganzen positiven Dezimalzahl in eine Zahl mit Basis **b** (2 <= **b** <= 36) soll als Algorithmus in Form eines **Programmablaufplanes** formuliert werden. Im Algorithmus sollen die Dezimalzahl **d** und die Basis **b** eingelesen und auf **d >= 0** und 2 <= **b** <= 36 getestet werden. Die Speicherung der Ziffern der umgewandelten Zahl soll in einem Vektor in Form von ASCII-Zeichen erfolgen. Der Vektor soll ausgegeben werden.

5. Der Algorithmus der **4.Aufgabe** ist als **C-Programm** zu implementieren. Die Umrechnung von **d** in eine Zahl zur Basis **b** soll über eine Funktion mit 3 Parametern (**unsigned long d, unsigned long b, char *z**) und dem Funktionstyp **int** erfolgen. Im Falle einer unzulässigen Basis (b<2 oder b>36) wird eine **1** zurückgegeben, sonst **0**. Das Ergebnis der Umwandlung wird als Folge von ASCII-Zeichen in **z** eingetragen. Eine Funktion **reverse(char *)** kehrt den Inhalt von **z** (vor der Ausgabe) um.

6. Die Umwandlung einer gebrochenen positiven Dezimalzahl der Form **0.f..f** in eine Zahl mitBasis **b** (2 <= **b** <= 36) soll als Algorithmus formuliert werden. Im Algorithmus sollen der Bruch **d** , die Basis**b** und die Anzahl der Nachkommastellen im Zielsystem eingelesen und auf **d >= 0** und 2 <= **b** <= 36 getestet werden. Die Speicherung der Ziffern der umgewandelten Zahl soll in einem Vektor in Form von ASCII-Zeichen erfolgen. Der Vektor soll ausgegeben werden.

7. Der Algorithmus der 6. Aufgabe soll analog zur 5.Aufgabe als C-Programm implementiert werden. Beispiel für den Dialog:

```
Umwandlung eines Dezimalbruchs in beliebiges System
===================================================
Umzuwandelnder Dezimalbruch aus [0,1] ? 0.125
Basis des Zielsystems aus [2,36] ? 16
Anzahl der Nachkommastellen im Zielsystem ? 10
(0.125)10  =  (0.2)16
          0.125 * 16 =            2 Ueberlauf  2
              0 * 16 =            0 Ueberlauf  0
```