



Halle, 20. November 2003

**Rechnerarchitektur und Rechnerorganisation** (WS 2003/04)  
Übungsserie 6

**Aufgabe 6.1.** (4 Punkte)

Gegeben sei eine Menge  $M = \{w_1, \dots, w_8\}$  von acht Elementen.

- Geben Sie einen Code  $c$  fester Länge an, der alle Elemente aus  $M$  codiert! Dieser Code soll 3-fehlererkennend aber nicht 4-fehlererkennend sein, wobei Sie die Länge des Codes selbst festlegen können.
- Zeigen Sie, daß Ihr Code  $c$  diese Eigenschaften erfüllt!
- Für welches  $k$  ist Ihr Code  $k$ -fehlerkorrigierend und nicht mehr  $(k+1)$ -fehlerkorrigierend? Begründen Sie Ihre Aussage!

— **Aufgabe 6.2.** (4 Punkte)

Zeigen Sie, daß für eine 2-dimensionale Paritätsüberprüfung gilt: die Parität der Zeilen-Paritäts-Bits ist gleich der Parität der Spalten-Paritäts-Bits.

**Aufgabe 6.3.** (10 Punkte)

Erweitern/Ändern Sie das Programm aus der Serie 5, um folgende Funktionalität zu erreichen:

Das Programm soll wie bisher innerhalb einer Schleife jeweils eine Zeichenkette mit maximal 10 Zeichen vom Nutzer erfragen.

Die Zeichenkette ist ebenfalls wie bisher zu interpretieren, also auf die Eingabe von:

- Zahlen (ohne Vorzeichen) mit 1 bis 10 Stellen
- oder Operationssymbole  $+$ ,  $-$ ,  $*$  oder  $/$
- oder das Zeichen 'q'

entsprechend zu reagieren. Die Verarbeitung dieser Eingaben soll auch wie in der Serie 5 erfolgen.

Zusätzlich soll nun das Programm auf die **Eingabe des Zeichens 'h'** reagieren, indem es folgende Operation ausführt:

Alle auf dem Stack nach vorherigen Eingaben und Berechnungen verbleibenden Zahlen sollen als die Menge der Codewörter  $c(A)$  eines hier nicht näher erläuterten Codes  $c : A \rightarrow \{0, 1\}^{32}$  fester Länge aufgefaßt werden. Es gibt keine weiteren Codeworte!

Berechnen und geben Sie die Distanz des Codes  $c$  aus. Überlegen Sie sich hierzu, wie Sie die Distanz zwischen zwei Codewörtern berechnen können und darauf basierend einen effektiven Algorithmus, um die Distanz des Codes zu bestimmen.

Nach der Ausgabe der Distanz als Reaktion auf die Eingabe des Zeichens 'h' soll das Programm mit dem nächsten Schleifendurchlauf fortfahren. Der Stack ist vorher zu leeren, insofern dies nicht bei der Berechnung der Distanz bereits erfolgte!

Überlegen Sie sich, welche Fehleingaben möglich sind, z.B. wenn die Elemente auf dem Stack keine Codewörter sein können oder der Stack leer ist. Geben Sie in diesem Fall eine passende Fehlermeldung aus, leeren Sie den Stack und setzen Sie mit dem nächsten Schleifendurchlauf fort.