Prof. Franz Schweiggert   Abteilung Angewandte Informationsverarbeitung
Christian Ehrhardt

Allgemeine Informatik 3 (WS 2002/2003)

Abgabetermin 11.9. 2002

Kompression (10 Punkte)

Wir haben ja bereits gesehen, daß Buchstaben in einem Rechner als Zahl bestehend aus 8 Bits dargestellt werden. Das hat aber den Nachteil, daß jedes Zeichen genauso viel Platz braucht, egal wie häufig es in einer durchschnittlichen Datei vorkommt. Das wollen wir jetzt ändern. Die sechs Buchstaben “e”, “r”, “n”, “s”, “t” und “l” sind im deutschen recht häufig, außerdem werden wohl auch das Leerzeichen und das Newline Zeichen recht oft vorkommen. Diese Zeichen sollen also wenig Platz benötigen. Also definieren wir zunächst, daß diese wie folgt als Bitfolge von je 4 Bits kodiert werden:

e	=	1000	r	=	1001	n	=	1010	s	=	1011
t	=	1100	l	=	1101	Leerzeichen	=	1110	Newline	=	1111

Am nächsthäufigsten kommen dann wohl die restlichen Kleinbuchstaben vor. Diese sollen durch 7 Bits kodiert werden. Die ersten beiden Bits sollen immer 01 sein, die übrigen 5 Bits geben die laufende Nummer des Kleinbuchstabens im Alphabet an, beginnend bei 0. Die restlichen Möglichkeiten von 7 Bit Zahlen, die mit 01 beginnen werden wie in der Tabelle angegeben durch Satzzeichen aufgefüllt:

a	=	01 00000	b	=	01 00001	c	=	01 00010	d	=	01 00011
e	=	01 00100	f	=	01 00101	g	=	01 00110	h	=	01 00111
...		...	z	=	01 11001	.	=	01 11010	,	=	01 11011
;	=	01 11100	:	=	01 11101	(	=	01 11110	)	=	01 11111

Das Leerzeichen zwischen den Bits dient hier nur der Lesbarkeit. Alle übrigen Zeichen werden durch 10 Bits kodiert, indem den 8 Bits Ihres ASCII-Codes die zwei Bits “00” vorangestellt werden, also etwa A = 00 01000001 (wobei dezimal 65 oder binär 01000001 dem ASCII-Code des Zeichens A entspricht.

Auf diese Weise läßt sich jeder Text als eine Folge von einzelnen Bits schreiben. Um den kodierten Text besser lesbar zu halten werden jeweils 16 aufeinanderfolgende Bits zu einer Dezimalzahl zusammengefaßt. Aufeinanderfolgende Zahlen werden durch Leerzeichen oder Newlines getrennt. Falls beim kodieren am Ende des Bitstromes nicht genau 16 Bit für die letzte Zahl übrig bleiben wird mit Einsen aufgefüllt. Beim dekodieren erscheinen diese Einsen ev. als zusätzlichen Newlines am Ende.

Eure Aufgabe ist es jetzt, ein Programm zum DEKODIEREN eines auf diese Weise kodierten

Textes zu schreiben. Dabei soll von der Standardeingabe gelesen und auf die Standardausgabe geschrieben werden. Falls beim Dekodieren am Ende einige Bits übrig bleiben, die kein vollständiges Zeichen mehr ergeben, sollen diese einfach ignoriert werden.

Kodierungs Beispiel

Der Text “10 Punkte” wird wie folgt kodiert:

Buchstabe	1	0	Leerzeichen	P	u
Kodierung	00 00110001	00 00110000	1110	00 01010000	01 10100
Buchstabe	n	k	t	e	
Kodierung	1010	01 01010	1100	1000	

Als Bitstrom in 16er Gruppen betrachtet ergibt das:

```
0000110001000011 0000111000010100 0001101001010010 101011001000
```

Oder mit Einsen aufgefüllt:

```
0000110001000011 0000111000010100 0001101001010010 1010110010001111
```

Interpretiert man diese Gruppen von 16 Bits jetzt als Integerzahl, so ergibt sich folgende Ausgabe des Kodierungsprogramms:

```
3139 3604 6738 44175
```

Dekodierung

Um eine Folge von Bits wieder dekodieren zu können ist es wichtig, sich eine Eigenschaft der beschriebenen Kodierung klarzumachen: Man weiß zu Beginn nicht, wieviele Bits zum nächsten Zeichen gehören, es könnten ja 4, 7 oder 10 sein. Um herauszubekommen, wieviele Bits zum nächsten Zeichen gehören muß man also zunächst das erste Bit betrachten. Handelt es sich um eine 1, dann weiß man, daß 4 Bits zu diesem Zeichen gehören, weil alle anderen kodierten Zeichen mit einer 0 beginnen würden. War das erste Bit dagegen eine 0, so muß noch das zweite Bit betrachtet werden um zu entscheiden, ob das Zeichen mit 7 oder mit 10 Bit kodiert wurde.
Bemerkung: Allgemein kann ein nach diesem Muster kodierter Bitstrom wieder dekodiert werden, wenn es kein Zeichen gibt, dessen (komplette) Kodierung als Folge von Bits der Beginn der Kodierung eines anderen Zeichens ist. Diese Eigenschaft wird Präfixfreiheit genannt.