

# LZW-Codierung (1)

- Tabelle (tab) mit Abbildung Zeichenreihe -> Indizes
  - Vorbesetzung z.B. ASCII-Zeichen -> ASCII-Code  
(muß nicht explizit gespeichert und übertragen werden)
- Prinzipieller Ablauf:

```
SeqChar p := < input.getChar() >;  
Char k := input.getChar();  
repeat  
    if tab.contains(p & < k >  
        then p := p & < k >  
        else int c = tab.newIndex(p & <k>);  
            tab.put(p & < k >, c);  
            output.write(tab.getIndex(p));  
            p := < k >;  
    endif  
    Char k := input.getChar();  
until k = EOF  
output.write(tab.getIndex(p));
```

# LZW-Codierung (2)

- Vorbesezte Tabelle (z.B.):  
[(, 97), (**>**, 98), (, 99), (, 100), (, 101), (, 102), (, 103),  
(, 104), (, 105), (, 106), (, 107), (, 108), (, 109),  
(, 110), (, 111), (, 112), (, 113), (, 114), (, 115),  
(, 116), (, 117), (, 118), (, 119), (, 120), (, 121),  
(, 122)]
- Für neue Einträge z.B. Nummern von 255 aufwärts verwendet.

# LZW-Codierung (3)

- Beispieltext: "bananenanbau"
- Ablauf:

Lesen (k)	Codetabelle schreiben (p & <k>)	Ausgabe	Puffer füllen (p)
b			<b>
a	(<ba>, 256)	98	<a>
n	(<an>, 257)	97	<n>
a	(<na>, 258)	110	<a>
n			<an>
e	(<ane>, 259)	257	<e>
n	(<en>, 260)	101	<n>
a			<na>
n	(<nan>, 261)	258	<n>
b	(<nb>, 262)	110	<b>
a			<ba>
u	(<bau>, 263)	256	<u>
EOF		117	

# LZW-Decodierung (2)

- Prinzipieller Algorithmus:

```
SeqChar p := <>;
k := input.getCode();
output.write(tab.getWord(k));
old := k;
while k ≠ EOF do
    k := input.getCode();
    q := firstChar(tab.getWord(k));
    output.write(tab.getWord(k));
    p := tab.getWord(old);
    int c = tab.newIndex(p & <q>);
    tab.put(q, c);
    old := k;
enddo;
```

*Hinweis:* Hier ist ein Spezialfall nicht berücksichtigt, in dem der Eintrag nicht in der Tabelle vorhanden sein kann. Vollständiger Algorithmus sh. z.B. Henning-Buch.

# LZW-Decodierung (3)

- Beispielcode: "98-97-110-257-101-258-110-256-117"
- Ablauf:

Lesen (k)	Ausgabe (q ist jeweils unterstrichen)	Puffer füllen (p)	Codetabelle schreiben (p & <q>)	Merken (old)
98	b			98
97	<u>a</u>	b	(<ba>, 256)	97
110	<u>n</u>	a	(<an>, 257)	110
257	<u>an</u>	n	(<na>, 258)	257
101	<u>e</u>	an	(<ane>, 259)	101
258	<u>na</u>	e	(<en>, 260)	258
110	<u>n</u>	na	(<nan>, 261)	110
256	<u>ba</u>	n	(<nb>, 262)	256
117	<u>u</u>	ba	(<bau>, 263)	117
EOF				