

$\Sigma = \{0, 1, \dots, 63\}$
alábec

LZW-COMPRESS(In, Out)

$D := \{\sigma_1, \dots, \sigma_d\}; \text{ kód} := d+1$	
$s := \text{get_char}(In)$	
$\neg \text{eof}(In)$	
$c := \text{get_char}(In)$	
$(\sigma_c := -) \in D$	
$s := \sigma_c$	$\text{write}(Out, \text{kód}(D, s))$
$D := DU\{\sigma_c; \text{kód}\}^*$	$D := DU\{\sigma_c; \text{kód}\}^*$
$s := c; \text{kód}++$	
$\text{write}(Out, \text{kód}(D, s))$	

$\text{kód}(D, s) = a$ D szótárban az s string kódja

σ_i a σ_i betűkölle string

$\text{eof}(In) \Leftrightarrow$ az In inputnál \forall karaktert \neg beolvastunk

" σ_i " olyan rendezett pár, aminek az első komponense σ_i

* Az új jött utasítások csak akkor használandók, ha $\text{kód} \leq \text{MAXKÓD}$, ahol MAXKÓD a kódok ~~maximális~~ előre rögzített hosszát jel. Pl. ha ez 12 bit, akkor $\text{MAXKÓD} = 2^{12} - 1 = 4095$

LZW-DECOMPRESS(In, Out, Σ)

$D := \{\sigma_1, \dots, \sigma_d\}; \text{ kód} := d+1$	
$k := \text{get_code}(In)$	
$s := \text{string}(D, k)$	
$\text{write}(Out, s)$	
$\neg \text{eof}(In)$	
$k := \text{get_code}(In)$	
$(-; k) \in D$	
$t := \text{string}(D, k)$	$t := s + k$
$\text{write}(Out, t)$	$\text{write}(Out, t)$
$D := DU\{\sigma_t; \text{kód}\}^*$	$D := DU\{t; \text{kód}\}^*$
$s := t; \text{kód}++$	

" $-; k$ " olyan rendezett pár, aminek a 2. komponense k

Uj: A fenti elágazásban az " $(-; k) \in D$ " feltétel helyén " $k < \text{kód}$ " is állhatna (ingyessé lehet tenni az előzőt)

* Itt $k = \text{kód}$ teljesül.

s, t : string
 $k, \text{kód}$: kódok
 t_1 : a t string első betűje
 $\text{string}(D, k) =$
 a D szótárban a k kódhoz megfelelő string
 $\text{eof}(In) \Leftrightarrow$
 az In inputnál \forall karaktert beolvastunk