

# Vektoren als endliche Abbildungen

## Ziel

Für endliche Menge  $K$  von *Schlüsseln* jeweils jedem  $k \in K$  einen Wert  $f(k) \in V$  zuordnen.

## Prinzip

Hierfür Bijektion  $\pi: K \rightarrow \{0, \dots, |K| - 1\}$  verwenden um Schlüssel auf Indizes abzubilden.

Dann Abspeichern der  $f(k)$  in einem Vektor  $v$  als  $v_{\pi(k)}$ .

## Hier konkret

$K$  alle Strings der Länge drei, Indizes  $0 \dots 17575$  mit z.B.

$$ABC \hat{=} 0 \cdot 26^2 + 1 \cdot 26^1 + 2 \cdot 26^0$$

# Hinweise zur Programmierung

- ▶ Einlesen von Standardeingabe aus Datei (statt langem Tippen)

Mit Datei `ciphertext.txt` und Programm `a.out`:

```
./a.out < ciphertext.txt
```

- ▶ `string encrypt(string key, string plaintext)`
- ▶ `string decrypt(string key, string ciphertext)`
- ▶ **Automatische Typumwandlung** `char`  $\leftrightarrow$  `int` anhand ASCII-Tabelle
- ▶ `'D' - 'A' == 3`
- ▶ `int ord(char c) { return c-'A'; }`

	30	40	50	60	70	80	90	100	110	120
	-----									
0:	(	2	<	F	P	Z	d	n	x	
1:	)	3	=	G	Q	[	e	o	y	
2:	*	4	>	H	R	\	f	p	z	
3:	!	+	?	I	S	]	g	q	{	
4:	"	,	@	J	T	^	h	r		
5:	#	-	7	A	K	U	_	i	s	}
6:	\$	.	8	B	L	V	`	j	t	~
7:	%	/	9	C	M	W	a	k	u	
8:	&	0	:	D	N	X	b	l	v	
9:	'	1	;	E	O	Y	c	m	w	

Abbildung: ASCII