

## 1.3 Kody alfanumeryczne

### 1. Uwagi wstępne

Jeśli zbiorem obiektów kodowanych  $V_1$  jest zbiór znaków to kod  $f: V_1 \rightarrow V_2^*$  (dla ustalonego alfabetu  $V_2$  na ogół binarnego) nazywamy kodem alfanumerycznym (ang. alphanumeric code).

W tym podrozdziale poznamy szereg binarnych kodów alfanumerycznych. Najpopularniejsze binarne kody alfanumeryczne to ośmiobitowy kod ASCII (American Standard Code for Information Interchange) i szesnastobitowy kod Unicode. Są to oczywiście kody o stałej długości słowa kodowego.

Znany każdemu telegrafistcie i harcerzowi kod Morse'a jest również kodem alfanumerycznym binarnym ale kodem zmiennej długości. Kody alfanumeryczne ASCII zmiennej długości mają specyficzne zalety opisane bliżej w podrozdziale o kompresji danych.

Kod ASCII został wprowadzony w USA w 1963 roku jako kod 7 bitowy. Kod ASCII w swej podstawowej wersji jest więc w zasadzie 7-mio bitowym kodem ale uzupełnionym z reguły bitem parzystości. Jako kod 8 bitowy ma jednak szereg odmian tzw. wersji narodowych.

Kod ASCII jest równoważny z kodem ISO-7 (ISO to skrót od International Organization for Standardization). Kod ISO-7 został opisany w normie ISO 646 wydanej w roku 1973. Norma ISO 646 przewiduje wprowadzenie znaków narodowych co wiąże się z rozszerzeniem kodu ASCII do kodu 8 bitowego

ISO-8859-2 to ogólnie już przyjęty standard kodowania polskich liter wg polskich norm PN stanowiący rozszerzenie ASCII.

Kod alfanumeryczny UNICODE jest standardem ISO/IEC-10646. Jest to kod 16 bitowy (pierwsza wersja) lub 31 bitowy (druga wersja). UNICODE umożliwia zapisanie wszystkich znaków z alfabetów narodowych (również cyrylicy, alfabetu chińskiego i alfabetu japońskiego)

Kod ASCII w wersji podstawowej koduje 128 znaków. Pierwsze 33 znaki (uporządkowane wg wartości w kodzie NKB słowa kodowego to tzw. znaki sterujące takie jak CR (Carriage Return czyli „powrót karetki”) lub LF (Line Feed czyli „od nowego wiersza”). Służą one do sterowania systemem drukowania lub wyświetlania znaków. Pozostałe znaki to m.in. małe i duże litery alfabetu angielskiego, cyfry, znaki przestankowe oraz kilka symboli matematycznych takich jak nawiasy i znak równości.

### 2. Kod ASCII

W technice cyfrowej i wywodzącej się z niej inżynierii komputerowej dominują obecnie kody alfanumeryczne oparte na standardzie ASCII (American Standard Code for Information Interchange), znanym także jako ANSI X3.4 oraz ISO-646-US i US-ASCII. W standardzie tym 128 podstawowym znakom, takim jak litery alfabetu łacińskiego, cyfry, znaki przestankowe oraz kody sterujące, przyporządkowano liczby z zakresu od 0 do 127. Z uwagi na oczywiste problemy z sortowaniem tak różnorodnych znaków tabele kodowe zestawia się według wartości kodów (dziesiętnych i szesnastkowych) - jak w poniższej tabeli.

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	NUL <i>null</i>	32	20	space	64	40	@	96	60	`
1	01	SOH <i>start of heading</i>	33	21	!	65	41	A	97	61	a
2	02	STX <i>start of text</i>	34	22	"	66	42	B	98	62	b
3	03	ETX <i>end of text</i>	35	23	#	67	43	C	99	63	c
4	04	EOT <i>end of transmission</i>	36	24	\$	68	44	D	100	64	d
5	05	ENQ <i>enquiry</i>	37	25	%	69	45	E	101	65	e
6	06	ACK <i>acknowledge</i>	38	26	&	70	46	F	102	66	f
7	07	BEL <i>bell</i>	39	27	'	71	47	G	103	67	g
8	08	BS <i>backspace</i>	40	28	(	72	48	H	104	68	h
9	09	TAB <i>horizontal tabulation</i>	41	29	)	73	49	I	105	69	i
10	0A	LF <i>line feed</i>	42	2A	*	74	4A	J	106	6A	j
11	0B	VT <i>vertical tabulation</i>	43	2B	+	75	4B	K	107	6B	k
12	0C	FF <i>form feed</i>	44	2C	,	76	4C	L	108	6C	l
13	0D	CR <i>carriage return</i>	45	2D	-	77	4D	M	109	6D	m
14	0E	SO <i>shift out</i>	46	2E	.	78	4E	N	110	6E	n
15	0F	SI <i>shift in</i>	47	2F	/	79	4F	O	111	6F	o
16	10	DLE <i>data link escape</i>	48	30	0	80	50	P	112	70	p
17	11	DC1 <i>device control one</i>	49	31	1	81	51	Q	113	71	q
18	12	DC2 <i>device control two</i>	50	32	2	82	52	R	114	72	r
19	13	DC3 <i>device control three</i>	51	33	3	83	53	S	115	73	s
20	14	DC4 <i>device control four</i>	52	34	4	84	54	T	116	74	t
21	15	NAK <i>negative acknowledge</i>	53	35	5	85	55	U	117	75	u
22	16	SYN <i>synchronous idle</i>	54	36	6	86	56	V	118	76	v
23	17	ETB <i>end of transmission block</i>	55	37	7	87	57	W	119	77	w
24	18	CAN <i>cancel</i>	56	38	8	88	58	X	120	78	x
25	19	EM <i>end of medium</i>	57	39	9	89	59	Y	121	79	y
26	1A	SUB <i>substitute</i>	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC <i>escape</i>	59	3B	;	91	5B	[	123	7B	{
28	1C	FS <i>file separator</i>	60	3C	<	92	5C	\	124	7C	
29	1D	GS <i>group separator</i>	61	3D	=	93	5D	]	125	7D	}
30	1E	RS <i>record separator</i>	62	3E	>	94	5E	^	126	7E	~
31	1F	US <i>unit separator</i>	63	3F	?	95	5F	_	127	7F	DEL

Tab. 1. US-ASCII

ASCII jest kodem 7-bitowym, przy czym oryginalny standard nie definiuje roli ósmego bitu, który w czysto sprzętowych realizacjach cyfrowych układów sterowania ciągle bywa wykorzystywany do kontroli parzystości lub do przechowywania dodatkowego atrybutu (np. podświetlenia). Najczęściej jednak ósmy bit służy rozszerzeniu podstawowego kodu ASCII o niezbędne znaki alfabetów narodowych, symbole matematyczne, znaki semigraficzne itp.. Niegdyś próbowano dopasować kod ASCII do warunków lokalnych poprzez odmienne wykorzystanie niektórych wartości kodów; np. w ISO-646-DE pod wartościami z zakresu 123-126 (7B-7E) kryją się odpowiednio znaki ä, ö, ü i ß. W miarę, jak rosły potrzeby, pojawiały się kolejne kody o rozmaitych sposobach i zakresach implementacji dodatkowych znaków, a wielu producentów sprzętu i oprogramowania wprowadzało własne "standardy". Z tych najbardziej znane jest rozszerzenie kodu ASCII wprowadzone przez firmę IBM, czyli CP437 (IBM PC Extended ASCII czyli DosLatinUS),

które zainicjowało całą serię tzw. stron kodowych, począwszy od CP850 (DosLatin1) i CP852 (DosLatin2), obecnych w kartach graficznych, drukarkach i w MS-DOS.

	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F
70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F
80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F
A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF

Tab. 2. CP437

80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F
A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF

Tab. 3. CP850

80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F
A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF

Tab. 4. CP852

ISO 8859 to zestaw kilkunastu tabel wykorzystujących wszystkie 256 wartości 8-bitowego słowa kodowego i rozszerzających właściwą tabelę ASCII (kody od 0 do 127 pozostają bez zmian) o znaki pochodzące z poszczególnych regionów Europy, alfabet grecki, cyrylicę

itp. Podobnie jak w podstawowym kodzie ASCII, pierwsze 32 pozycje każdej rozszerzonej tabeli ISO 8859-*n* to niedrukowalne znaki sterujące. Odpowiednikiem CP850 jest ISO 8859-1, czyli Latin1, dedykowany Europie Zachodniej. Odpowiednikiem CP852 jest ISO 8859-2 (Latin2) zawierający komplet polskich znaków, a np. greckie litery można znaleźć w ISO 8859-7 (Greek)

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
128	80	PAD <i>padding character</i>	160	A0		192	C0	Ř	224	E0	í
129	81	HOP <i>high octet preset</i>	161	A1	Ą	193	C1	Á	225	E1	á
130	82	BPH <i>break permitted here</i>	162	A2	˘	194	C2	Â	226	E2	â
131	83	NBH <i>no break here</i>	163	A3	Ł	195	C3	Ã	227	E3	ã
132	84	IND <i>index</i>	164	A4	◻	196	C4	Ä	228	E4	ä
133	85	NEL <i>next line</i>	165	A5	Ĺ	197	C5	Í	229	E5	í
134	86	SSA <i>start of selected area</i>	166	A6	Š	198	C6	Ć	230	E6	ć
135	87	ESA <i>end of selected area</i>	167	A7	§	199	C7	Ç	231	E7	ç
136	88	HTS <i>character tabulation set</i>	168	A8	¨	200	C8	Č	232	E8	č
137	89	HTJ <i>character tabulation with justification</i>	169	A9	Š	201	C9	É	233	E9	é
138	8A	VTB <i>line tabulation set</i>	170	AA	Ş	202	CA	È	234	EA	è
139	8B	PLD <i>partial line forward</i>	171	AB	Ť	203	CB	Ě	235	EB	ě
140	8C	PLU <i>partial line backward</i>	172	AC	Ž	204	CC	Ě	236	EC	ě
141	8D	RI <i>reverse line feed</i>	173	AD	<i>soft hyphen</i>	205	CD	Í	237	ED	í
142	8E	SS2 <i>single-shift two</i>	174	AE	Ž	206	CE	Î	238	EE	î
143	8F	SS3 <i>single-shift three</i>	175	AF	Ž	207	CF	Ď	239	EF	ď
144	90	DCS <i>device control string</i>	176	B0	◦	208	D0	Đ	240	F0	đ
145	91	PU1 <i>private use one</i>	177	B1	ą	209	D1	Ň	241	F1	ň
146	92	PU2 <i>private use two</i>	178	B2	˘	210	D2	Ň	242	F2	ň
147	93	STS <i>set transmit state</i>	179	B3	ł	211	D3	Ó	243	F3	ó
148	94	CCH <i>cancel character</i>	180	B4	´	212	D4	Ô	244	F4	ô
149	95	MW <i>message waiting</i>	181	B5	ł	213	D5	Õ	245	F5	õ
150	96	SPA <i>start of guarded area</i>	182	B6	ś	214	D6	Ö	246	F6	ö
151	97	EPA <i>end of guarded area</i>	183	B7	˘	215	D7	×	247	F7	÷
152	98	SOS <i>start of string</i>	184	B8	,	216	D8	Ř	248	F8	ř
153	99	SGCI <i>single graphic character introducer</i>	185	B9	š	217	D9	Ů	249	F9	ů
154	9A	SCI <i>single character introducer</i>	186	BA	ş	218	DA	Ú	250	FA	ú
155	9B	CSI <i>control sequence introducer</i>	187	BB	ť	219	DB	Ů	251	FB	ů
156	9C	ST <i>string terminator</i>	188	BC	ž	220	DC	Ü	252	FC	ü
157	9D	OSC <i>operating system command</i>	189	BD	¨	221	DD	Ý	253	FD	ý
158	9E	PM <i>privacy message</i>	190	BE	ž	222	DE	Ť	254	FE	ť
159	9F	APC <i>application program command</i>	191	BF	ž	223	DF	ß	255	FF	·

Tab. 5. ISO 8859-2 (Latin2)

Wprowadzenie standardu ISO 8859 miało w założeniu uporządkować zasady kodowania rozszerzeń ASCII. Niestety, nie tylko nie zapobiegło definitywnie dalszemu mnożeniu się stosowanych kodów alfanumerycznych, lecz wręcz zapoczątkowało kolejną ich falę,

tak w ramach samego standardu (wystarczy prześledzić ewolucję obsługi języków krajów nadbałtyckich od ISO 8859-4 (Latin4) poprzez ISO 8859-10 (Latin6) do ISO 8859-13 (Latin7 Baltic Rim)), jak i incjatyw zewnętrznych (na przykład Microsoft zmodyfikował tabelę ISO 8859-2 i wprowadził ją do Windows jako stronę kodową CP1250 (WinLatin2)). Główną tego przyczyną są obiektywne ograniczenia wynikające ze zbyt małej ilości słów kodu 8-bitowego.

80	€		,		,,	...	†	‡		‰	Š	‹	Š	Ť	Ž	Ž																
	91	¢	92	93	„	94	95	96	97		99	9A	9B	9C	9D	9E	9F															
						•	-	-		™	Š	›	Š	ť	ž	ž																
A0	A1	˘	A2	˘	A3	Ł	Ŧ	À	Ì	8	..	A9	©	A8	§	A6	«		AD	-	AE	®	AF	˙								
E0	°		B1	±		B2	˘	Ł	˘	B5	μ	B6	¶	B7	•	B8	˘	B9	q	\$	»	Ł	˘	BE	İ	BF	˙					
C0	Ř	C1	Ā	C2	Ā	C3	Ā	C4	Ā	C5	Ĺ	C6	Ĉ	C7	Ĉ	C8	Ĉ	C9	Ė	Ė	Ė	Ė	Ė	Ė	Ė	Ė	Ė					
D0	Đ	D1	Ñ	D2	Ñ	D3	Ó	D4	Ô	D5	Õ	D6	Ö	D7	×	D8	Ř	D9	Ů	DA	Ú	DB	Ů	DC	Ü	DD	Ý	DE	Ť	DF	ß	
E0	ř	E1	ā	E2	ā	E3	ā	E4	ā	E5	ĺ	E6	ĉ	E7	ĉ	E8	ĉ	E9	ė	EA	e	EB	ė	EC	ė	ED	ı	EE	ı	EF	˘	d
F0	đ	F1	ñ	F2	ñ	F3	ó	F4	ô	F5	õ	F6	ö	F7	÷	F8	ř	F9	ů	FA	ú	FB	ů	FC	ü	FD	ý	FE	ı	FF	.	

**Tab. 6. CP1250 (WinLatin2)**

### 3. Unicode

W zapisie UTF-8 każdy znak UCS/Unicode jest przedstawiany w postaci sekwencji od jednego do sześciu 8-bitowych bajtów, zależnie od wartości samego znaku. Poniższa tabela obrazuje zasadę, na jakiej to się odbywa:

Liczba bajtów	1. bajt	2. bajt	3. bajt	4. bajt	Liczba bitów	Maksymalna wartość
UTF-8					Unicode	Hex Dec
1	0xxxxxxx				7	7F 127
2	110xxxxx	10xxxxxx			11	7FF 2047
3	1110xxxx	10xxxxxx	10xxxxxx		16	FFFF 65535
4	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx	21	1FFFFF 2097151
5	111110xx	10xxxxxx	itd.		26	3FFFFFFF 67108863
6	1111110x	10xxxxxx	itd.		31	7FFFFFFF 2147483647

**Tab. 7. UTF-8**

```
0xxxxxxx : pierwszy i jedyny bajt sekwencji
10xxxxxx : drugi i dalsze bajty sekwencji
110xxxxx : pierwszy bajt sekwencji 2-bajtowej
1110xxxx : pierwszy bajt sekwencji 3-bajtowej
11110xxx : pierwszy bajt sekwencji 4-bajtowej
111110xx : pierwszy bajt sekwencji 5-bajtowej
1111110x : pierwszy bajt sekwencji 6-bajtowej
```

**Tab. 8. Znaczenie bajtu w zapisie UTF-8**

W wielobajtowej sekwencji bity oznaczone 'x'-ami czytane od 1-szego, najstarszego bajtu tworzą właściwą wartość znaku UCS/Unicode. Z kolei wartość bieżącego bajtu wskazuje

na jego miejsce w sekwencji UTF-8. W ten sposób zapis UTF-8 jest kompatybilny z 7-bitowym US-ASCII i zachowuje względną zwartość tekstów o niewielu znakach rozszerzonych, pozwalając jednocześnie na zapis nawet 31-bitowych wartości i na łatwą do realizacji synchronizację i interpretację przetwarzanych sekwencji.

**Przykład:** Oto kilka przykładów 16-bitowych wartości Unicode w zapisie UTF-8:

Unicode	UTF-8
Hex	Hex
0001	01
007F	7F
0080	C2 80
07FF	DF BF
0800	E0 A0 80
0FFF	E0 BF BF
1000	E1 80 80
FFFF	EF BF BF ■

Zapis UTF-8 jest nadmiarowy, bowiem np. wartościom 16-bitowym przyporządkowuje wartości 24-bitowe (3-bajtowe). W efekcie istnieje wiele ciągów bajtów, które nie są legalnymi sekwencjami UTF-8, nawet jeżeli wykonanie przekształcenia odwrotnego jest technicznie możliwe.

**Przykład:** Ciąg bajtów C2 00 nie jest sekwencją UTF-8, ponieważ jego drugi bajt nie jest zgodny z wzorcem 10xxxxxx.

**Przykład:** Ciąg bajtów C0 80 nie jest legalną sekwencją UTF-8, ponieważ jest dłuższy, niż trzeba: po zastosowaniu do niego maski binarnej 110xxxxx 10xxxxxx uzyskujemy wynik 00000 000000, czyli zerową wartość kodu, którą w UTF-8 zapisuje się poprawnie w jednym bajcie.

sekwencja	C0 80	11000000	10000000
maska		110xxxxx	10xxxxxx
wynik	00	00000	000000 ■

Oprócz zapisu UTF-8 spotyka się czasem UTF-7 wykorzystujący tylko 7 bitów używanego słowa oraz UTF-16 będący po prostu zapisem kolejnych kodów.

A oto reprezentacja polskich „ogonków” w Unicode:

Znak	Kod	Znak	Kod
ą	0105	Ą	0104
ć	0107	Ć	0106
ę	0119	Ę	0118
ł	0142	Ł	0141
ń	0144	Ń	0143
ó	00F3	Ó	00D3
ś	015B	Ś	015A
ż	017A	Ż	0179
ź	017C	Ź	017B

**Tab. 9. Polskie znaki w Unicode**