Wykład 5: Łańcuchy znakowe.

dr inż. Andrzej Stafiniak

Wrocław 2023





Podstawowe informacje o łańcuchach znakowych

- Znak symbol graficzny którego znaczenie interpretowane jest w zdefiniowany sposób.
- Pogrupowane sekwencje znaków tworzą programy źródłowe, którymi możemy sterować pracę komputera. Łańcuch to zespół znaków traktowana jako całość.
- W języku C brak jest specjalnego typu danych do reprezentacji całych napisów, dlatego napis jest traktowany jako łańcuch znaków.
- W języku C istnieje typ znakowy char. Zmienna tego typu może przechowywać wartość całkowitą (char litera = 97) jak i odpowiadający tej wartości znak (char litera = 'a'). Znakom przypisane są wartości liczbowe zgodnie z tabelą ASCII.
- Typ char zajmuje 1 bajt czyli 8 bitów. Standardowy kod ASCII potrzebuje
 7 bitów (0-127). Rozszerzona tablica ASCII 8 bitów (0-255).



Podstawowa tablica ASCII

Bin	Dec	Hex	Znak	Skrót
0000 0000	0	00	Null	NUL
0000 0001	1	01	Start of Heading	SOH
0000 0010	2	02	Start of Text	STX
0000 0011	3	03	End of Text	ETX
0000 0100	4	04	End of Transmission	EOT
0000 0101	5	05	Enquiry	ENQ
0000 0110	6	06	Acknowledge	ACK
0000 0111	7	07	Bell	BEL
0000 1000	8	08	Backspace	BS
0000 1001	9	09	Horizontal Tab	HT
0000 1010	10	0A	Line Feed	LF
0000 1011	11	0B	Vertical Tab	VT
0000 1100	12	0C	Form Feed	FF
0000 1101	13	0D	Carriage Return	CR
0000 1110	14	0E	Shift Out	SO
0000 1111	15	0F	Shift In	SI
0001 0000	16	10	Data Link Escape	DLE
0001 0001	17	11	Device Control 1 (XON)	DC1
0001 0010	18	12	Device Control 2	DC2
0001 0011	19	13	Device Control 3 (XOFF)	DC3
0001 0100	20	14	Device Control 4	DC4
0001 0101	21	15	Negative Acknowledge	NAK
0001 0110	22	16	Synchronous Idle	SYN
0001 0111	23	17	End of Transmission Block	ETB
0001 1000	24	18	Cancel	CAN
0001 1001	25	19	End of Medium	EM
0001 1010	26	1A	Substitute	SUB
0001 1011	27	1B	Escape	ESC
0001 1100	28	1C	File Separator	FS
0001 1101	29	1D	Group Separator	GS
0001 1110	30	1E	Record Separator	RS
0001 1111	31	1F	Unit Separator	US

Bin	Dec	Hex	Znak	E
0100 0000	64	40	@	0110
0100 0001	65	41	Α	0110
0100 0010	66	42	В	0110
0100 0011	67	43	С	0110
0100 0100	68	44	D	0110
0100 0101	69	45	Е	0110
0100 0110	70	46	F	0110
0100 0111	71	47	G	0110
0100 1000	72	48	Н	0110
0100 1001	73	49	- 1	0110
0100 1010	74	4A	J	0110
0100 1011	75	4B	K	0110
0100 1100	76	4C	L	0110
0100 1101	77	4D	М	0110
0100 1110	78	4E	N	0110
0100 1111	79	4F	0	0110
0101 0000	80	50	Р	0111
0101 0001	81	51	Q	0111
0101 0010	82	52	R	0111
0101 0011	83	53	S	0111
0101 0100	84	54	Т	0111
0101 0101	85	55	U	0111
0101 0110	86	56	V	0111
0101 0111	87	57	W	0111
0101 1000	88	58	X	0111
0101 1001	89	59	Υ	0111
0101 1010	90	5A	Z	0111
0101 1011	91	5B	[0111
0101 1100	92	5C	1	0111
0101 1101	93	5D]	0111
0101 1110	94	5E	۸	0111
				1

Bin	Dec	Hex	Znak	Skrót
0110 0000	96	60	,	
0110 0001	97	61	а	
0110 0010	98	62	b	
0110 0011	99	63	С	
0110 0100	100	64	d	
0110 0101	101	65	е	
0110 0110	102	66	f	
0110 0111	103	67	g	
0110 1000	104	68	h	
0110 1001	105	69	i	
0110 1010	106	6A	j	
0110 1011	107	6B	k	
0110 1100	108	6C	- 1	
0110 1101	109	6D	m	
0110 1110	110	6E	n	
0110 1111	111	6F	0	
0111 0000	112	70	р	
0111 0001	113	71	q	
0111 0010	114	72	r	
0111 0011	115	73	S	
0111 0100	116	74	t	
0111 0101	117	75	u	
0111 0110	118	76	٧	
0111 0111	119	77	W	
0111 1000	120	78	х	
0111 1001	121	79	у	
0111 1010	122	7A	Z	
0111 1011	123	7B	{	
0111 1100	124	7C	- 1	
0111 1101	125	7D	}	
0111 1110	126	7E	~	
0444 4444	407	75	Delete	DEL

Podstawowa tablica ASCII + Windows-1250 (CP-1250)

Bin	Dec	Hex	Znak	Skrót	Bin	Dec	Hex	Znak	Bin	Dec	Hex	Znak	Bin	Dec	Hex	Znak	Skrót
0000 0000	0	00	Null	NUL	0010 0000	32	20	Spacja	0100 0000	64	40	@	0110 0000	96	60	,	
0000 0001	1	01	Start of Heading	SOH	0010 0001	33	21	!	0100 0001	65	41	Α	0110 0001	97	61	а	
0000 0010	2	02	Start of Text	STX	0010 0010	34	22	"	0100 0010	66	42	В	0110 0010	98	62	b	
0000 0011	3	03	End of Text	ETX	0010 0011	35	23	#	0100 0011	67	43	С	0110 0011	99	63	С	
0000 0100	4	04	End of Transmission	EOT	0010 0100	36	24	\$	0100 0100	68	44	D	0110 0100	100	64	d	
0000 0101	5	05	Enquiry	ENQ	0010 0101	37	25	%	0100 0101	69	45	Е	0110 0101	101	65	е	
0000 0110	6	06	Acknowledge	ACK	0010 0110	38	26	&	0100 0110	70	46	F	0110 0110	102	66	f	
0000 0111	7	07	Bell	BEL	0010 0111	39	27	1	0100 0111	71	47	G	0110 0111	103	67	g	
0000 1000	8	08	Backspace	BS	0010 1000	40	28	(0100 1000	72	48	Н	0110 1000	104	68	h	
0000 1001	9	09	Horizontal Tab	HT	0010 1001	41	29)	0100 1001	73	49	- 1	0110 1001	105	69	i	
0000 1010	10	0A	Line Feed	LF	0010 1010	42	2A	*	0100 1010	74	4A	J	0110 1010	106	6A	j	
0000 1011	11	0B	Vertical Tab	VT	0010 1011	43	2B	+	0100 1011	75	4B	K	0110 1011	107	6B	k	
0000 1100	12	0C	Form Feed	FF	0010 1100	44	2C	,	0100 1100	76	4C	L	0110 1100	108	6C	- 1	
0000 1101	13	0D	Carriage Return	CR	0010 1101	45	2D	-	0100 1101	77	4D	М	0110 1101	109	6D	m	
0000 1110	14	0E	Shift Out	so	0010 1110	46	2E		0100 1110	78	4E	N	0110 1110	110	6E	n	
0000 1111	15	0F	Shift In	SI	0010 1111	47	2F	1	0100 1111	79	4F	0	0110 1111	111	6F	0	
0001 0000	16	10	Data Link Escape	DLE	0011 0000	48	30	0	0101 0000	80	50	Р	0111 0000	112	70	р	
0001 0001	17	11	Device Control 1 (XON)	DC1	0011 0001	49	31	1	0101 0001	81	51	Q	0111 0001	113	71	q	
0001 0010	18	12	Device Control 2	DC2	0011 0010	50	32	2	0101 0010	82	52	R	0111 0010	114	72	r	
0001 0011	19	13	Device Control 3 (XOFF)	DC3	0011 0011	51	33	3	0101 0011	83	53	S	0111 0011	115	73	s	
0001 0100	20	14	Device Control 4	DC4	0011 0100	52	34	4	0101 0100	84	54	Т	0111 0100	116	74	t	
0001 0101	21	15	Negative Acknowledge	NAK	0011 0101	53	35	5	0101 0101	85	55	U	0111 0101	117	75	u	
0001 0110	22	16	Synchronous Idle	SYN	0011 0110	54	36	6	0101 0110	86	56	V	0111 0110	118	76	V	
0001 0111	23	17	End of Transmission Block	ETB	0011 0111	55	37	7	0101 0111	87	57	W	0111 0111	119	77	w	
0001 1000	24	18	Cancel	CAN	0011 1000	56	38	8	0101 1000	88	58	X	0111 1000	120	78	X	
0001 1001	25	19	End of Medium	EM	0011 1001	57	39	9	0101 1001	89	59	Υ	0111 1001	121	79	у	
0001 1010	26	1A	Substitute	SUB	0011 1010	58	ЗА	:	0101 1010	90	5A	Z	0111 1010	122	7A	z	
0001 1011	27	1B	Escape	ESC	0011 1011	59	3B	;	0101 1011	91	5B	[0111 1011	123	7B	{	
0001 1100	28	1C	File Separator	FS	0011 1100	60	3C	<	0101 1100	92	5C	١	0111 1100	124	7C	- 1	
0001 1101	29	1D	Group Separator	GS	0011 1101	61	3D	=	0101 1101	93	5D]	0111 1101	125	7D	}	
0001 1110	30	1E	Record Separator	RS	0011 1110	62	3E	>	0101 1110	94	5E	٨	0111 1110	126	7E	~	
0001 1111	31	1F	Unit Separator	US	0011 1111	63	3F	?	0101 1111	95	5F	_	0111 1111	127	7F	Delete	DEL

					Dována	ادماد				
Hex	CP-1250	ISO 8859-2	Hex	CP-1250	Porównani ISO 8859-2	Нех	CP-1250 ISO 8859-	2 Hex	CP-1250	ISO 8859-2
0x80	€	ZK	0xA0	GF-1230	130 0039-2	0xC0	Ŕ	0xE0	GF-1230	f 0009-2
0x81	NZ	ZK	0xA0	v	Α	0xC1	Á	0xE1		á
0x82	,	ZK	0xA2		7	0xC2	Â	0xE2		â
0x83	, NZ	ZK	0xA3		Ł	0xC3	Ă	0xE3		ă
0x84	,,,	ZK	0xA4		п	0xC4	Ä	0xE4		ä
0x85		ZK	0xA5	Ą	Ľ	0xC5	Ĺ	0xE5		i
0x86	t	ZK	0xA6	-	Ś	0xC6	Ċ	0xE6		ć
0x87	‡	ZK	0xA7		§	0xC7	Ç	0xE7		ç
0x88	NZ	ZK	0xA8		-	0xC8	Č	0xE8		č
0x89	‰	ZK	0xA9	©	Š	0xC9	É	0xE9		ė
0x8A	Š	ZK	0xAA		Ş	0xCA	Ę	0xEA		ę
0x8B	(ZK	0xAB	«	Ť	0xCB	Ë	0xEB		ë
0x8C	Ś	ZK	0xAC	7	Ż	0xCC	Ě	0xEC		ě
0x8D	Ť	ZK	0xAD			0xCD	ĺ	0xED		Í
0x8E	Ž	ZK	0xAE	®	Ž	0xCE	Î	0xEE		î
0x8F	Ż	ZK	0xAF		Ż	0xCF	Ď	0xEF		ď
0x90	NZ	ZK	0xB0		۰	0xD0	Ð	0xF0		đ
0x91		ZK	0xB1	±	ą	0xD1	Ń	0xF1		ń
0x92	,	ZK	0xB2		c.	0xD2	Ň	0xF2		ň
0x93	и	ZK	0xB3		ł	0xD3	Ó	0xF3		ó
0x94	,,	ZK	0xB4			0xD4	Ô	0xF4		Ô
0x95	•	ZK	0xB5	μ	ľ	0xD5	Ő	0xF5		ő
0x96	-	ZK	0xB6	¶	Ś	0xD6	Ö	0xF6		ö
0x97	_	ZK	0xB7		ŭ	0xD7	×	0xF7		÷
0x98	NZ	ZK	0xB8			0xD8	Ř	0xF8		ř
0x99	TM	ZK	0xB9	ą	š	0xD9	Ů	0xF9		ů
0x9A	š	ZK	0xBA		ş	0xDA	Ú	0xFA		ú
0x9B)	ZK	0xBB	»	ť	0xDB	Ü	0xFB		ű
0x9C	ś	ZK	0xBC	Ľ	Ź	0xDC	Ü	0xFC		ü
0x9D	ť	ZK	0xBD		*	0xDD	Ý	0xFD		ý
0x9E	Ž	ZK	0xBE	ľ	ž	0xDE	Ţ	0xFE		ţ
0x9F	Ż	ZK	0xBF		Ż	0xDF	ß	0xFF		

Windows-1250 – 8 bitowe rozszerzenie kodu ASCII zwane stronami kodowymi. Używane są przez system Windows do reprezentacji tekstów w językach środkowoeuropejskich.



Podstawowa tablica ASCII + Windows-1250 (CP-1250)

Bin	Dec	Hex	Znak	Skrót	Bin	Dec	Hex	Znak	Bin	Dec	Hex	Znak	Bin	Dec	Hex	Znak	Skrót
0000 0000	0	00	Null	NUL	0010 0000	32	20	Spacja	0100 0000	64	40	@	0110 0000	96	60		
0000 0001	1	01	Start of Heading	SOH	0010 0001	33	21	!	0100 0001	65	41	Α	0110 0001	97	61	а	
0000 0010	2	02	Start of Text	STX	0010 0010	34	22	"	0100 0010	66	42	В	0110 0010	98	62	b	
0000 0011	3	03	End of Text	ETX	0010 0011	35	23	#	0100 0011	67	43	С	0110 0011	99	63	С	
0000 0100	4	04	End of Transmission	EOT	0010 0100	36	24	\$	0100 0100	68	44	D	0110 0100	100	64	d	
0000 0101	5	05	Enquiry	ENQ	0010 0101	37	25	%	0100 0101	69	45	Е	0110 0101	101	65	е	
0000 0110	6	06	Acknowledge	ACK	0010 0110	38	26	&	0100 0110	70	46	F	0110 0110	102	66	f	
0000 0111	7	07	Bell	BEL	0010 0111	39	27		0100 0111	71	47	G	0110 0111	103	67	g	
0000 1000	8	08	Backspace	BS	0010 1000	40	28	(0100 1000	72	48	Н	0110 1000	104	68	h	
0000 1001	9	09	Horizontal Tab	нт	0010 1001	41	29)	0100 1001	73	49	- 1	0110 1001	105	69	i	
0000 1010	10	0A	Line Feed	LF	0010 1010	42	2A	*	0100 1010	74	4A	J	0110 1010	106	6A	j	
0000 1011	11	0B	Vertical Tab	VT	0010 1011	43	2B	+	0100 1011	75	4B	К	0110 1011	107	6B	k	
0000 1100	12	0C	Form Feed	FF	0010 1100	44	2C	,	0100 1100	76	4C	L	0110 1100	108	6C	I	
0000 1101	13	0D	Carriage Return	CR	0010 1101	45	2D	-	0100 1101	77	4D	М	0110 1101	109	6D	m	
0000 1110	14	0E	Shift Out	so	0010 1110	46	2E		0100 1110	78	4E	N	0110 1110	110	6E	n	
0000 1111	15	0F	Shift In	SI	0010 1111	47	2F	1	0100 1111	79	4F	0	0110 1111	111	6F	0	
0001 0000	16	10	Data Link Escape	DLE	0011 0000	48	30	0	0101 0000	80	50	Р	0111 0000	112	70	р	
0001 0001	17	11	Device Control 1 (XON)	DC1	0011 0001	49	31	1	0101 0001	81	51	Q	0111 0001	113	71	q	
0001 0010	18	12	Device Control 2	DC2	0011 0010	50	32	2	0101 0010	82	52	R	0111 0010	114	72	r	
0001 0011	19	13	Device Control 3 (XOFF)	DC3	0011 0011	51	33	3	0101 0011	83	53	S	0111 0011	115	73	s	
0001 0100	20	14	Device Control 4	DC4	0011 0100	52	34	4	0101 0100	84	54	Т	0111 0100	116	74	t	
0001 0101	21	15	Negative Acknowledge	NAK	0011 0101	53	35	5	0101 0101	85	55	U	0111 0101	117	75	u	
0001 0110	22	16	Synchronous Idle	SYN	0011 0110	54	36	6	0101 0110	86	56	V	0111 0110	118	76	٧	
0001 0111	23	17	End of Transmission Block	ETB	0011 0111	55	37	7	0101 0111	87	57	W	0111 0111	119	77	W	
0001 1000	24	18	Cancel	CAN	0011 1000	56	38	8	0101 1000	88	58	X	0111 1000	120	78	х	
0001 1001	25	19	End of Medium	EM	0011 1001	57	39	9	0101 1001	89	59	Υ	0111 1001	121	79	у	
0001 1010	26	1A	Substitute	SUB	0011 1010	58	3A	:	0101 1010	90	5A	Z	0111 1010	122	7A	Z	
0001 1011	27	1B	Escape	ESC	0011 1011	59	3B	;	0101 1011	91	5B	[0111 1011	123	7B	{	
0001 1100	28	1C	File Separator	FS	0011 1100	60	3C	<	0101 1100	92	5C	1	0111 1100	124	7C	- 1	
0001 1101	29	1D	Group Separator	GS	0011 1101	61	3D	=	0101 1101	93	5D	1	0111 1101	125	7D	}	
0001 1110	30	1E	Record Separator	RS	0011 1110	62	3E	>	0101 1110	94	5E	٨	0111 1110	126	7E	~	
0001 1111	31	1F	Unit Separator	US	0011 1111	63	3F	?	0101 1111	95	5F	_	0111 1111	127	7F	Delete	DEL

Hex CP-1250 ISO 8859-2 ISO 8859-2 Hex CP-1250 ISO 8859-2 ISO 8859-						_ /						
0x80 € ZK 0xA0 0xC0 R 0xE0 f 0x81 WZ ZK 0xA1 A 0xC1 A 0xE1 a 0x82 ZK 0xA2 C 0xC2 A 0xE2 a 0x83 WZ ZK 0xA3 L 0xC3 A 0xE3 a 0x84 ZK 0xA4 B 0xC4 A 0xE4 a 0x85 ZK 0xA5 A L' 0xC5 L 0xE5 i 0x86 † ZK 0xA6 † Ś 0xC6 C 0xE6 C 0x86 † ZK 0xA6 † Ś 0xC7 Ç 0xE7 Ç 0x88 WZ ZK 0xA8 Š 0xC9 E 0xE9 E 0x88 ¢ ZK 0xAA Ş 0xCA E 0xEA Q <												
0.881 WZ ZK 0.x41 T A 0xC1 A 0xE1 â 0.x82 . ZK 0xA2 T 0xC2 A 0xE2 â 0.x83 WZ ZK 0xA3 L 0xC3 A 0xE3 â 0.x84 . ZK 0xA4 a 0xC4 Â 0xE4 â 0.x85 . . ZK 0xA5 A L' 0xC5 L 0xE6 c 0xE7 ç 0xE8 c 0xE8 c 0xE8 c 0xE8 c 0xE8 c 0xE8 c 0xE8					CP-1250	ISO 8859-2		CP-1250			CP-1250	
0.882 . ZK 0.xA2 - 0.xC2 Å 0.xE2 Å 0.883 MZ ZK 0.xA3 L 0.xC3 Å 0.xE3 Å 0.x84 - ZK 0.xA4 B 0.xC4 Å 0.xE4 Å 0.x85 ZK 0.xA5 Å L' 0.xC5 L 0.xE5 I 0.x86 † ZK 0.xA6 ‡ \$ 0.xC6 C 0.xE5 I 0.x86 † ZK 0.xA6 ‡ \$ 0.xC7 Ç 0.xE7 Ç 0.x88 MZ ZK 0.xA9 © \$ 0.xC9 E 0.xE9 è 0.XE9 0.XE9 0.XE9 i 0.XE9 0.XE9 <t< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></t<>												
0x83 N72 ZK 0xA3 L 0xC3 Å 0xE4 ä 0x84 ZK 0xA4 □ 0xC4 Å 0xE4 ä 0x85 ZK 0xA5 Å L 0xC5 L 0xE5 I 0x86 † ZK 0xA6 Å L 0xC6 C 0xE6 C 0x86 † ZK 0xA7 § 0xC7 Ç 0xE7 Ç 0x88 NZ ZK 0xA8 ¬ 0xC8 C 0xE8 C 0x89 ½ 0xAA § 0xCA Ę 0xEA ę 0x80 Ž 0xAA § 0xCA Ę 0xEB ē 0x80 † ZK 0xAD 0xCD İ 0xEC ē 0x80 † ZK 0xAB § Ž 0xCE Ī 0xEE Ī		NZ			ĭ	Ą						á
0x84 ZK 0xA4 □ 0xC4 Å 0xE4 å 0x85 ZK 0xA5 A L 0xC5 L 0xE5 I 0x86 † ZK 0xA6 ¦ Š 0xC6 C 0xE6 C 0x87 ‡ ZK 0xA7 § 0xC7 Ç 0xE7 Ç 0x88 M² ZK 0xA8 ¬ 0xC8 C 0xE8 c 0x88 M² ZK 0xA9 © Š 0xC9 È 0xE9 è 0x88 ½ ZK 0xAA § 0xCA Ę 0xE9 è 0x88 ¼ ZK 0xAB « † 0xCB Ē 0xEB è 0x80 † 0xAD ¬ Ż 0xCC Ē 0xEC ē 0x80 † 0xCD QXCD QXED QXED </td <td>0x82</td> <td>,</td> <td>ZK</td> <td>0xA2</td> <td></td> <td></td> <td>0xC2</td> <td></td> <td>Â</td> <td>0xE2</td> <td></td> <td>â</td>	0x82	,	ZK	0xA2			0xC2		Â	0xE2		â
0x85 ZK 0xA5 A L 0xC5 L 0xE5 I 0x86 ↑ ZK 0xA6 ↓ \$ 0xC6 C 0xE6 6 0x87 ‡ ZK 0xA7 § 0xC7 Ç 0xE7 Ç 0x88 NZ ZK 0xA8 ¬ 0xC8 Ĉ 0xE8 ĉ 0x89 ‰ ZK 0xA9 ⊚ Š 0xC9 Ê 0xE9 ê 0x80 Š ZK 0xAA § 0xCA Ę 0xEA ę 0x80 Š ZK 0xAA § 0xCB Ê 0xEB e 0x80 Š ZK 0xAD ¬ Ž 0xCC Ê 0xEC e 0x80 Ť 0xAD 0xCD I 0xED I 0xEE I 0x81 Ž 0xAE Ž 0xCF D	0x83	NZ	ZK	0xA3		Ł	0xC3		Ä	0xE3		ă
0x86 ↑ ZK 0xA6 ↓ \$ 0xC6 C 0xE7 Ç 0x87 ‡ ZK 0xA7 § 0xC7 Ç 0xE7 Ç 0x88 NZ ZK 0xA8 ¬ 0xC8 Ç 0xE8 Ç 0x89 % ZK 0xA9 © Š 0xC9 Ê 0xE9 ê 0x80 Š ZK 0xAA § 0xCA Ę 0xEA ę 0x88 ⟨ ZK 0xAB « ↑ 0xCB Ê 0xEB e 0x80 ↑ ZK 0xAD 0xCD ↓ 0xEC e 0xEC e 0xEC e 0xEC e 0xEC e 0xEC ↓ 0xEC </td <td>0x84</td> <td>,</td> <td>ZK</td> <td>0xA4</td> <td></td> <td>n</td> <td>0xC4</td> <td></td> <td>Ä</td> <td>0xE4</td> <td></td> <td>ä</td>	0x84	,	ZK	0xA4		n	0xC4		Ä	0xE4		ä
0x87 ‡ ZK 0xA7 § 0xC7 Ç 0xE7 Ç 0x88 NZ ZK 0xA8 " 0xC8 Ĉ 0xE8 Ĉ 0x89 ‰ ZK 0xA9 © Š 0xC9 Ê 0xE9 ê 0x8A Š ZK 0xAA Ş 0xCA E 0xEA ę 0x8B ⟨ ZK 0xAB « Ť 0xCB Ê 0xEB ê 0x8C Ś ZK 0xAD 0xCD Ĭ 0xEC ê 0x8D Ť ZK 0xAD 0xCD Ĭ 0xED Ĭ 0x8E Ž ZK 0xAE ® Ž 0xCE Ĭ 0xEE Ĭ 0x8E Ž ZK 0xAE ® Ž 0xCF Ď 0xEF ď 0x90 NZ ZK 0xB0 * 0xD0 D 0xF0	0x85		ZK	0xA5	Ą	Ľ	0xC5		Ĺ	0xE5		Í
0x88 NZ ZK 0xA8 " OxC8 C OxE8 6 0x89 ‰ ZK 0xA9 © S 0xC9 É 0xE9 è 0x8A Š ZK 0xAA Ş 0xCA E 0xEA ę 0x8B C ZK 0xAA Ş 0xCB E 0xEB ë 0x8C Š ZK 0xAC ¬ 2 0xCB E 0xEC è 0x8D T ZK 0xAD 0xCD I 0xED I 0x8E Z ZK 0xAE ® 2 0xCE I 0xEE I 0x8E Z ZK 0xAE ® 2 0xCE I 0xEE I 0x90 NZ ZK 0xB ® 2 0xD 0xD 0xEF d 0x91 ZK 0xB 0xB 0xD 0xD 0xF0 d 0x92 ZK 0xB 0xB 0xD 0xD 0xF2 ñ 0x93 ZK 0xB 0xB 0xD 0xD 0xF2 ñ 0x94 ZK 0xB I 0xD	0x86	†	ZK	0xA6	- 1	Ś	0xC6		Ċ	0xE6		Ć
0x89 ‰ ZK 0xA9 ⊚ Š 0xC9 É 0xE9 é 0x8A Ŝ ZK 0xAA Ş 0xCA E 0xEA ę 0x8B ⟨ ZK 0xAB « Ť 0xCB É 0xEB ē 0x8C Ŝ ZK 0xAB « Ť 0xCC É 0xEC é 0x8D Ť ZK 0xAD 0xCD I 0xED I 0x8E Ž ZK 0xAE ® Ž 0xCE I 0xEE I 0x8F Ž ZK 0xAE Z 0xCF D 0xEE I 0x99 M7Z ZK 0xB0 * 0xD0 D 0xF0 d D 0xF0 d D 0xF1 ñ 0xF1 ñ 0xF1 ñ 0xF2 ñ 0xF2 ñ 0xF2 ñ 0xF2 ñ	0x87	‡	ZK	0xA7		§	0xC7		Ç	0xE7		ç
0x8A \$ ZK 0xAA \$ 0xCA E 0xEA € 0x8B ⟨ ZK 0xAB ⟨ T 0xCB E 0xEB E 0x8C \$ ZK 0xAC ¬ Z 0xCC E 0xEC E 0x8D T ZK 0xAD 0xCD I 0xED I 0x8E Z ZK 0xAD 0xCD I 0xED I 0x8E Z ZK 0xAE ® Z 0xCE I 0xEE I 0x8F Z ZK 0xAF Z 0xCF D 0xEF d 0x99 I ZK 0xB0 P 0xDD D 0xF0 d 0x91 I ZK 0xB1 ± q 0xD1 N 0xF1 ñ 0xF2 ñ 0xF2 ñ 0xF2 ñ 0xF2 ñ 0xF2	0x88	NZ	ZK	0xA8		-	0xC8		Č	0xE8		Č
0x8B ⟨ ZK 0xAB ⟨ † 0xCB € 0xEB € 0x8C \$ ZK 0xAC ¬ Ż 0xCC Ê 0xEC € 0x8D † ZK 0xAD QxCD I 0xED I 0x8D † 0xCD I 0xED I 0xED I 0x8E Ž 0xAE ® Ž 0xCF Ď 0xEE I 0x8D Ž 0xCF Ď 0xEF ď Ø	0x89	‰	ZK	0xA9	©	Š	0xC9		É	0xE9		ė
0x8C Ś ZK 0xAC ¬ Ż 0xCC Ē 0xEC Ē 0x8D † ZK 0xAD 0xCD İ 0xED Ĭ 0x8E Ż ZK 0xAE ® Ž 0xCE Ĭ 0xEE Ĭ 0x8F Ż ZK 0xAF Ž 0xCF Ď 0xEE Ĭ 0x90 MZ XK 0xB0 * 0xD0 D 0xF0 d 0x91 ZK 0xB0 * 0xD0 D 0xF0 d d 0x91 ZK 0xB1 ± q 0xD1 Ñ 0xF1 n 0xF1 n 0xF2 n </td <td>0x8A</td> <td>Š</td> <td>ZK</td> <td>0xAA</td> <td></td> <td>Ş</td> <td>0xCA</td> <td></td> <td>Ę</td> <td>0xEA</td> <td></td> <td>ę</td>	0x8A	Š	ZK	0xAA		Ş	0xCA		Ę	0xEA		ę
0x8D T ZK 0xAD 0xCD I 0xED I 0x8E Z ZK 0xAE ⊗ Z 0xCE I 0xEE I 0x8F Z ZK 0xAF Z 0xCF D 0xEE I 0x90 NZ ZK 0xB0 * 0xD0 D 0xF0 d 0x91 ZK 0xB1 ± q 0xD1 N 0xF1 n 0x92 ZK 0xB2 , 0xD2 N 0xF2 n 0x93 ZK 0xB2 , 0xD2 N 0xF2 n 0x93 ZK 0xB3 I 0xD3 O 0xF3 O 0x94 ZK 0xB4 I 0xD4 OxD4 OxF4 O 0x95 ZK 0xB5 µ I 0xD5 OxF5 O OxF5 O 0x96 ZK <td< td=""><td>0x8B</td><td>(</td><td>ZK</td><td>0xAB</td><td>«</td><td>Ť</td><td>0xCB</td><td></td><td>Ë</td><td>0xEB</td><td></td><td>ë</td></td<>	0x8B	(ZK	0xAB	«	Ť	0xCB		Ë	0xEB		ë
0x8E Ż ZK 0xAE ® Ż 0xCE I 0xEE I 0x8F Ż ZK 0xAF Ż 0xCF Ď 0xEF ď 0x90 MZ XK 0xB0 * 0xD0 Ð 0xF0 ď 0x91 * ZK 0xB1 ± ¶ 0xD1 Ñ 0xF1 ñ 0x92 * ZK 0xB2 * 0xD2 Ñ 0xF2 ñ 0x93 * ZK 0xB3 I 0xD3 Ó 0xF3 ó 0x93 * ZK 0xB4 * 0xD4 Ó 0xF4 ó 0x94 * ZK 0xB4 * 0xD4 Ó 0xF4 ó 0x95 * ZK 0xB6 ¶ f 0xD6 Ø 0xF5 ó 0x96 * ZK 0xB6 ¶ f 0xD7	0x8C	Ś	ZK	0xAC	7	Ź	0xCC		Ě	0xEC		ě
0x8F Ż ZK 0xAF Ż 0xCF Ď 0xEF ď 0x90 NZ ZK 0xB0 * 0xD0 Ď 0xF0 ď 0x91 * ZK 0xB1 ± q 0xD1 Ñ 0xF1 ñ 0x92 * ZK 0xB2 , 0xD2 Ñ 0xF2 ñ 0x93 * ZK 0xB3 i 0xD3 Ó 0xF3 ó 0x94 * ZK 0xB4 * 0xD4 Ó 0xF4 ó 0x95 * ZK 0xB4 * 0xD4 Ó 0xF4 ó 0x95 * ZK 0xB6 ¶ í oxD6 Ó 0xF5 ó 0x96 - ZK 0xB6 ¶ í oxD7 × 0xF7 + 0x98 * 0xB8 * 0xD8 Ñ 0xF8	0x8D	Ť	ZK	0xAD			0xCD		ĺ	0xED		Ĺ
0x90 NZ ZK 0xB0 " 0xD0 D 0xF0 d 0x91 . ZK 0xB1 ± q 0xD1 Ñ 0xF1 ñ 0x92 . ZK 0xB2 . 0xD2 Ñ 0xF2 ñ 0x93 . ZK 0xB3 i 0xD3 Ó 0xF3 ó 0xF3 ó 0x94 . ZK 0xB4 . 0xD4 Ó 0xF4 ó 0xF4 ó 0x95 . ZK 0xB5 µ r 0xD5 Ó 0xF5 ó 0xF5 ó 0x96 . ZK 0xB6 ¶ ś 0xD6 Ó 0xF6 ó 0xF6 ó 0x97 . ZK 0xB7 . . 0xD7 x 0xF7 + 0x98 . . 0xD8 R 0xF8 ñ 0xF7 + 0x99 . ZK 0xB9 q š 0xD9 Û 0xF9 û 0xF9 û 0x99 . ZK 0xB8 . 0xD8 Û 0xF6 û 0xF6	0x8E	Ž	ZK	0xAE	®	Ž	0xCE		Î	0xEE		î
0.090 1/2 2K 0xB0 0xD0 0xD0 0xF0 0xF0 0xF0 0xF0 0xF0 0xF0 0xF0 0xF1 n 0xF1 n 0xF1 n 0xF1 n 0xF1 n 0xF2 n 0xF2 n 0xF2 n 0xF2 n 0xF2 n 0xF2 n 0xF3 o 0xF3 o 0xF3 o 0xF3 o 0xF3 o 0xF4 o o 0xF4 o o 0xF4 o o 0xF4 o o oxF5 o o 0xF5 o o oxF5 o o oxF5 o o oxF6 o o oxF7 + o oxF7 + o oxF7 + o oxF7 o o oxF8 f o o o oxF8 f o o o o o o o o	0x8F	Ż	ZK	0xAF		Ż	0xCF		Ď	0xEF		ď
0x92 ∴ ZK 0xB2 ∴ 0xD2 Ñ 0xF2 ñ 0x93 " ZK 0xB3 I 0xD3 Ó 0xF3 ó 0x94 " ZK 0xB4 I 0xD4 Ó 0xF4 ó 0x95 • ZK 0xB5 µ I 0xD5 Ó 0xF5 ó 0x96 — ZK 0xB6 ¶ Í ś 0xD6 Ó 0xF6 ó 0x97 — ZK 0xB7 · ~ 0xD7 × 0xF7 + 0x98 WZ ZK 0xB8 . 0xD8 R 0xF8 f 0x99 ™ ZK 0xB9 q š 0xD9 Ú 0xF9 û 0x99 ¾ ZK 0xB8 % 0xDA Ú 0xFA û 0x99 ¾ ZK 0xBB %	0x90	NZ	ZK	0xB0		•	0xD0		Đ	0xF0		đ
0x92 ZK 0x82 0x02 N 0xF2 1 0x93 ZK 0xB3 I 0xD3 O 0xF3 O 0x94 ZK 0xB4 0xD4 O 0xF4 O 0x95 ZK 0xB5 µ r 0xD5 O 0xF5 O 0x96 ZK 0xB6 ¶ \$ 0xD6 O 0xF6 O 0x97 — ZK 0xB7 — 0xD7 × 0xF7 — 0x98 NZ ZK 0xB8 — 0xD8 R 0xF8 f 0x99 M ZK 0xB8 QXDA U 0xF9 U 0x9A \$ ZK 0xBB » t' 0xDB U 0xFB U 0x9B > ZK 0xBC L' ± 0xDC U 0xFC U 0x9D t' ZK 0xBD — 0xDD Y 0xFD Y	0x91	,	ZK	0xB1	±	ą	0xD1		Ń	0xF1		ń
0x95 ZK 0x85 0xD4 0xP4 0xP4 0 0x95 • ZK 0xB4 • 0xD5 0xP5 0 0xF5 0 0x95 • ZK 0xB6 ¶ \$ 0xD6 0xD6 0xF6 0 0x96 - ZK 0xB7 • 0xD7 × 0xF7 + 0x98 NZ 2K 0xB8 • 0xD8 0xD8 0xF8 † 0x99 ™ ZK 0xB9 q \$ 0xD9 Û 0xF9 û 0x9A \$ ZK 0xBA \$ 0xDA Û 0xFA û 0x9B > ZK 0xBB * '' 0xDB Û 0xFB û 0x9C \$ ZK 0xBC L' ½ 0xDC Û 0xFC û 0x9D '' ZK 0xBD • 0xDC Û 0xFD ý	0x92	,	ZK	0xB2			0xD2		Ň	0xF2		ň
0x95 ⋅ ∠K 0x85 μ Γ 0xD5 Ô 0xF5 ô 0x96 - ∠K 0x86 ¶ \$ 0xD6 Ô 0xF6 ô 0x97 - ∠K 0x87 · · 0xD7 × 0xF7 + 0x98 NZ 0xB8 . 0xD8 R 0xF8 f 0x99 ™ ZK 0xB9 q \$ 0xD9 Û 0xF9 û 0x9A \$ ZK 0xBA \$ 0xDA Û 0xFA û 0x9B > ZK 0xBB » t' 0xDB Û 0xFB û 0x9C \$ ZK 0xBC L' ½ 0xDC Û 0xFC û 0x9D t' ZK 0xBD . 0xDD Y 0xFD y	0x93	"	ZK	0xB3		ł	0xD3		Ó	0xF3		Ó
0x96 — ZK 0x86 ¶ \$ 0xD6 0 0xF6 6 0x97 — ZK 0x87 . . 0xD7 × 0xF7 + 0x98 NZ ZK 0x88 . 0xD8 R 0xF8 r̄ 0x99 ™ ZK 0xB9 q \$ 0xD9 Û 0xF9 Û 0x9A \$ ZK 0xBA \$ 0xDA Û 0xFA Û 0x9B > ZK 0xBB » t' 0xDB Û 0xFB Û 0x9C \$ ZK 0xBC L' ź 0xDC Û 0xFC Û 0x9D t' ZK 0xBD . 0xDD Y 0xFD Y	0x94	"	ZK	0xB4			0xD4		Ô	0xF4		ô
0x97 — ZK 0x87 · · 0xD7 × 0xF7 + 0x98 NZ ZK 0x88 , 0xD8 R 0xF8 f 0x99 ™ ZK 0xB9 q \$ 0xD9 Ú 0xF9 Ú 0x9A \$ ZK 0xBA \$ 0xDA Ú 0xFA Ú 0x9B > ZK 0xBB » t' 0xDB Ü 0xFC Ū 0x9D t' ZK 0xBD — 0xDD Y 0xFD Y	0x95	•	ZK	0xB5	μ	ľ	0xD5		Ö	0xF5		ő
0x98 NZ ZK 0xB8 . 0xD8 R 0xF8 f 0x99 ™ ZK 0xB9 q \$ 0xD9 Û 0xF9 Û 0x9A \$ ZK 0xBA \$ 0xDA Û 0xFA Û 0x9B > ZK 0xBB » t' 0xDB Û 0xFB Û 0x9C \$ ZK 0xBC L' ½ 0xDC Û 0xFC Û 0x9D t' ZK 0xBD T 0xDD Y 0xFD Y	0x96	-	ZK	0xB6	¶	Ś	0xD6		Ö	0xF6		ö
0X99 ™ ZK 0XB9 q \$ 0XD9 Ü 0XF9 Ü 0X9A \$ 0XBA \$ 0XDA Ü 0XFA Ü 0X9B > ZK 0XBB » t' 0XDB Ü 0XFB Ü 0X9C \$ ZK 0XBC L' ½ 0XDC Ü 0XFC Ü 0X9D t' ZK 0XBD V 0XDD Y 0XFD Y	0x97	_	ZK	0xB7		v	0xD7		×	0xF7		+
0x9A \$ ZK 0xBA \$ 0xDA Û 0xFA Û 0x9B > ZK 0xBB » t' 0xDB Û 0xFB Û 0x9C \$ ZK 0xBC L' 2 0xDC Û 0xFC Û 0x9D t' ZK 0xBD - 0xDD Y 0xFD Y	0x98	NZ	ZK	0xB8			0xD8		Ř	0xF8		ř
0x9B) ZK 0xBB » t' 0xDB Ü 0xFB Ü 0x9C \$ ZK 0xBC L' 2 0xDC Ü 0xFC Ü 0x9D t' ZK 0xBD - 0xDD Y 0xFD Y	0x99	TM	ZK	0xB9	ą	š	0xD9		Ů	0xF9		ů
0x9C \$ ZK 0xBC L' 2 0xDC Ü 0xFC Ü 0x9D t' ZK 0xBD - 0xDD Ŷ 0xFD Ŷ	0x9A	š	ZK	0xBA		ş	0xDA		Ú	0xFA		ú
0x9D t' Z K 0xBD - 0xDD Ý 0xFD ý	0x9B	>	ZK	0xBB	»	ť	0xDB		Ü	0xFB		ű
	0x9C	ś	ZK	0xBC	Ľ	Ź	0xDC		Ü	0xFC		ü
0x9E ž ZK 0xBE ľ ž 0xDE T 0xFE	0x9D	ť	ZK	0xBD			0xDD		Ý	0xFD		ý
	0x9E	ž	ZK	0xBE	r	ž	0xDE		Ţ	0xFE		ţ
0x9F	0x9F	Ź	ZK	0xBF		Ż	0xDF		ß	0xFF		

Alternatywa - zestaw znaków Unicode – system kodowania UTF-8, UTF-16, UTF-32



Łańcuchy znakowe

Znaki

np.: 'q' 'w' 'e' '!' '('

Łańcuch znaków

"Stop war!"

Stała łańcuchowa - #define NAPIS "Stop war!"

- W języku C brak jest typu danych do reprezentacji łańcuchów znaków.
- > Łańcuchy przechowywane są w tablicach zbudowanych z elementów typu char.

Tablica char'ów (ale jeszcze nie tablica zawierająca łańcuch znakowy)





Tablica

- > **Tablica** to ciąg wartości tego samego typu przechowywanych w kolejno po sobie następujących komórkach pamięci.
- Tak jak zmienne czy funkcję **tablice**, mają nazwę, czyli swój identyfikator.
- W celu dokonania deklaracji tablicy musimy użyć typu jaki będzie przechowywać tablica, nazwy tablicy oraz operatora indeksu [] wraz z rozmiarem tablicy. np.: char tab[9];
- Dostęp do składowych/elementów tablicy można uzyskać poprzez podanie indeksu tablicy – czyli odpowiedniego numeru komórki, pamiętając, że tablice numerowane są od ineksu 0.

komórka	S	t	0	р		w	а	r	!
indeks	0	1	2	3	4	5	6	7	8



Jednowymiarowa tablica char'ów

Składnia deklaracji statycznej tablicy jednowymiarowej jest następująca:

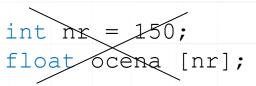
```
typ nazwaTablicy[rozmiar];
```

```
char tab[9]; int studentIndex[6]; float ocena[NR]
```

#define NR 150 char tab[9]; - deklaracja 9-cio elementowej tablicy char'ów

```
tab[0] = 'S';
```

tab[1] = 't'; Przypisanie wartości do poszczególnych komórek tablicy char'ów



komórka	S	t	0	р		w	а	r	- !
indeks	0	1	2	3	4	5	6	7	8

Dalej nie jest to tablica zawierająca łańcuch znakowy



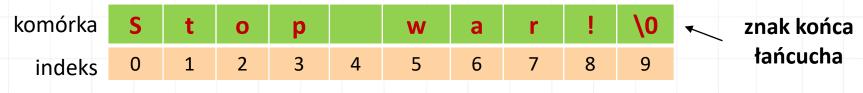
Inicjalizacja tablicy łańcuchem znakowym

- W celu możliwości użycia łańcuchów znakowych przez różne funkcje obsługi wejścia-wyjścia, w tablicy musi być przewidziane miejsce na specjalny znak końca łańcucha (znak zerowy) \0 null character
- > Do zainicjalizowania tablicy znakowej możemy wykorzystać następujące sposoby:

```
char tab[] = {'S', 't', 'o', 'p', ' ', 'w', 'a', 'r', '!', '\0'};

char tab[] = {"Stop war!"};

char tab[] = "Stop war!";
```

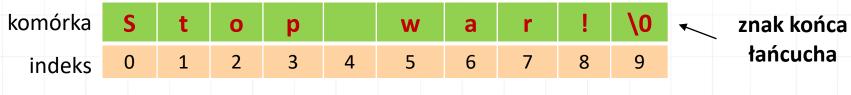


Tablica zawierająca łańcuch znakowy



Inicjalizacja tablicy łańcuchem znakowym

- Przedstawione sposoby deklaracji oraz inicjalizacji tablic nie wymagają podawania jawnie rozmiaru tablicy - kompilator zarezerwuje odpowiednią ilość komórek.
- Podając w sposób jawny rozmiar tablicy musimy pamiętać, że rozmiar musi być co najmniej o 1 większy niż ilość znaków.



Tablica zawierająca łańcuch znakowy



Inicjalizacja tablicy łańcuchem znakowym

```
#include <stdio.h>
                                                  Brak znaku zerowego \0
int main()
    char tab1 [] = {'H', 'e', 'l', 'l', 'o'};
    char tab2 [] = "Hello";
    printf("%s", tab1);
    printf(" - rozmiar tablicy: %d\n", sizeof tab1);
    printf("%s", tab2);
    printf(" - rozmiar tablicy: %d", sizeof tab2);
                                                               Przy braku znaku zerowego
    return 0;
                                                               funkcja printf() odwoła
                               Hello♀ a - rozmiar tablicy: 5
                                                               się do jakiejś przypadkowej
                               Hello - rozmiar tablicy: 6
                                                                                 wartości
     komórka
                                           W
                                            5
                0
                      1
                                3
                                      4
                                                  6
                                                                  9
       indeks
```

Funkcje obsługi znaków oraz łańcuchów znakowych

Plik nagłówkowy < ctype . h> - funkcje obsługi znaków

Funkcja sprawdza czy znak przekazany jako argument jest:

Nazwa funkcji	Opis					
isalnum	jest cyfrą lub literą alfabetu.					
isalpha jest literą alfabetu.						
iscntrl	jest znakiem kontrolnym.					
isdigit	jest cyfrą.					
isgraph	jest znakiem graficznym.					
islower jest małą literą alfabetu.						
isprint	jest znakiem drukowalnym.					
jest znakiem drukowalnym ale nie jest znakiem ispunct alfanumerycznym ani spacją.						
isspace	jest białym znakiem.					
isupper	jest dużą literą alfabetu.					
isxdigit	jest cyfrą szesnastkową.					
tolower	zwraca znak zamieniony z dużej litery na małą.					
toupper	zwraca znak zamieniony z małej litery na dużą.					

```
int isalpha(int ch);
```

funkcja ta zwraca wartość różną
 od zera gdy argument, który został
 przekazany do funkcji jest znakiem alfabetu.

```
char znak;
scanf("%c", &znak);
if(isalpha(znak))
  printf( ,,%c to litera
  alfabetu", znak);
else
  printf( ,,%c nie nalezy
  do alfabetu", znak);
```



Funkcje obsługi znaków oraz łańcuchów znakowych

Plik nagłówkowy < string.h> - funkcje obsługi łańcuchów znakowych

Nazwa funkcji	Opis
atol	Konwertuje wartość zapisaną w łańcuchu znaków do postaci liczby typu całkowitego (long).
strcat	Scala dwa łańcuchy znaków w jeden.
strchr	Szuka pierwszego wystąpienia znaku w łańcuchu znaków.
strcmp	Porównuje dwa łańcuchy znaków.
strcpy	Kopiuje łańcuch znaków do tablicy znaków.
strcspn	Szuka pierwszego wystąpienia znaku (z puli znaków) w łańcuchu znaków.
strerror	Zwraca łańcuch znaków zawierający komunikat błędu dla wskazanego numeru błędu.
stricmp	Porównuje dwa łańcuchy znaków (ignoruje wielość liter).
strlen	Oblicza długość łańcucha.
strncat	Scala dwa łańcuchy znaków w jeden. Uwzględnia maksymalną liczbę znaków, jaka może zostać dopisana.
strncmp	Porównuje określoną liczbę znaków dwóch łańcuchów znaków.
strncpy	Kopiuje określoną liczbę znaków łańcucha.
strrchr	Szuka ostatniego wystąpienia znaku w łańcuchu znaków.
strspn	Zwraca indeks pierwszego znaku, który nie należy do puli znaków.
strstr	Szuka pierwszego wystąpienia łańcucha znaków w innym łańcuchu znaków.
strtok	Zastępuje (w łańcuchu znaków) pierwszy znaleziony znak znakiem terminalnym.
strtol	Konwertuje wartość zapisaną w łańcuchu znaków w dowolnym systemie liczbowym do postaci liczby typu całkowitego (long).



strlen Oblicza długość łańcucha znaków.

Typ zwracany, liczby całkowite dodatnie unsigned int

Wskaźnik na pierwszy element tablicy w której przechowywany jest łańcuch

Funkcja zwraca ilość znaków w łańcuchu do napotkania znaku zerowego \0. W przypadku ciągów niezakończonych znakiem zerowym działanie funkcji jest nieprzewidziane.

```
char tab[] = "No war, no more!";
int counter;
counter = strlen (tab);
```

Jaka wartość zostanie przypisana do zmiennej counter?



Nagłówek funkcji size t strlen (const char *str)

```
strlen Oblicza długość łańcucha znaków.
```

```
#include <stdio.h>
#include <string.h>
int main()
                                                             Hello♀ a - dlugosci lancucha znakow: 8
                                                             Hello - dlugosci lancucha znakow: 5
    char tab1 [] = {'H', 'e', 'l', 'l', 'o'};
                                                              Hello - dlugosci lancucha znakow: 5
    char tab2 [] = "Hello";
    char tab3 [7] = "Hello";
    printf("\n%s", tab1);
    printf(" - dlugosci lancucha znakow: %d\n", strlen(tab1));
    printf("%s", tab2);
    printf(" - dlugosci lancucha znakow: %d\n", strlen(tab2));
    printf("%s", tab3);
    printf(" - dlugosci lancucha znakow: %d\n", strlen(tab3));
    return 0;
```

```
strcmpPorównuje dwa łańcuchy znaków.stricmpPorównuje dwa łańcuchy znaków (ignoruje wielość liter).strncmpPorównuje określoną liczbę znaków dwóch łańcuchów znaków.
```

```
int strcmp (const char *str1, const char *str2)
   Nagłówek funkcji
#include <stdio.h>
#include <string.h>
int main()
                                                      tab1: Hello♀a, tab2: Hello, tab3: Hello
                                                      strcmp (tab1, tab2): 1
   char tab1 [] = {'H', 'e', 'l', 'l', 'o'};
                                                      strcmp (tab2, tab3): 0
   char tab2 [] = "Hello";
   char tab3 [7] = "Hello";
   printf("\ntab1: %s, tab2: %s, tab3: %s\n", tab1, tab2, tab3);
   printf("strcmp (tab1, tab2): %d\n", strcmp(tab1,tab2));
   printf("strcmp (tab2, tab3): %d\n", strcmp(tab2,tab3));
   return 0;
                   Funkcja stremp przeszukuje łańcuchy do momentu znalezienia pierwszej pary
```

różnych znaków. Zwraca 1 - gdy pierwszy łańcuch jest mniejszy od drugiego oraz

-1 w przeciwnym wypadku. Gdy łańcuchy są takie same zwraca 0.

```
strcat
               Scala dwa łańcuchy znaków w jeden.
               Scala dwa łańcuchy znaków w jeden. Uwzględnia maksymalną liczbę znaków, jaka może zostać dopisana.
    strncat
                           char strcat (char *destination, const char *source)
   Nagłówek funkcji
                 char strncat (char *destination, const char *source, size t n)
#include <stdio.h>
#include <string.h>
                                     tab5: Tablica wynikowa musi byc zainicjalizowana oraz musi byc statycznie zdeklarowany rozmiar.
int main()
                                     tab6: Oraz mozna okreslic ile chcemy dolaczyc elementow do nowej Tablica wynikowa
    char tab2 [] = " musi";
   char tab3 [] = " byc";
    char tab4 []= " zainicjalizowana";
   char tab5 [100] = "Tablica wynikowa";
    char tab6 [100] = "Oraz mozna okreslic ile chcemy dolaczyc elementow do nowej ";
    strcat (tab5, tab2);
                                                        Do nas należy weryfikacja czy łańcuch dołączany
    strcat (tab5, tab3);
                                                                  zmieści się w tablicy pierwszego łańcucha.
    strcat (tab5, tab4);
    strcat (tab5, " oraz musi byc statycznie zdeklarowany rozmiar.");
    strncat (tab6, tab5, 16);
   printf("\ntab5: %s\n", tab5);
   printf("\ntab6: %s\n", tab6);
                                         Pierwszy znak drugiego łańcucha nadpisuje znak \0 pierwszego
                                         łańcucha. Na koniec nowego łańcucha dodawany jest znak zerowy.
   return 0;
```

```
strcpy Kopiuje łańcuch znaków do tablicy znaków.strncpy Kopiuje określoną liczbę znaków łańcucha.
```

```
Nagłówek funkcji char strcpy (char *destination, const char *source)

char strncpy (char *destination, const char *source, size_t n)
```

```
#include <stdio.h>
#include <string.h>

int main()
{
    char tab1 [] = "Tekst do kopiowania";
    char tab2 [100];
    char tab3 [100];

    strcpy (tab2, tab1);
    strcpy (tab3, "Inny tekst do kopiowania1");
    printf("\ntab2: %s\n", tab2);
    printf("\ntab3: %s\n", tab3);

    strncpy (tab3, tab1, 5);
    printf("\ntab3: %s\n", tab3);

    return 0;
```

- Funkcje strcpy i strncpy kopiują łańcuch source do łańcucha destination do momentu napotkania znaku \0 lub przekopiowania n znaków.
- ➤ Jeśli kopiowany łańcuch jest krótszy od n, to zostanie skopiowany cały ze znakiem \0. W przeciwnym wypadku łańcuch końcowy może nie mieć znaku zerowego!
- Do nas należy weryfikacja czy łańcuch kopiowany zmieści się w tablicy pierwszego łańcucha.

```
tab2: Tekst do kopiowania
tab3: Inny tekst do kopiowania1
tab3: Teksttekst do kopiowania1
```



strchr

Nagłówek funkcji

Szuka pierwszego wystąpienia znaku w łańcuchu znaków.

char *strchr (const char *str, int c)

```
#include <stdio.h>
#include <string.h>
int main()
    char polecenie [] = "Zachowaj odstęp!";
    char *str;
    int c = 'a';
    str = strchr(polecenie, c);
    printf("\nstr: %d\n", str);
    if (str!=NULL) {
        printf("\nstr: %d\n", *str);
        printf("\nn: %d\n", str-polecenie+1);
    printf("\nPrzeszukaj caly lancuch", str-polecenie+1);
    while (str!=NULL) {
        printf ("\nn: %d\n",str-polecenie+1);
        str=strchr(str+1, c);
    return 0;
```

Zwracany typ to wskaźnik czyli adres do znalezionego znaku

```
str: 6422279
str: 97
n: 2
Przeszukaj caly lancuch
n: 2
n: 7
```

strtol

Konwertuje wartość zapisaną w łańcuchu znaków w dowolnym systemie liczbowym do postaci liczby typu całkowitego (long).

```
long strtol(const char *str, char **end, int base)
  Nagłówek funkcji
#include <stdio.h>
#include <stdlib.h>
                                                                         Wskaźnik na wskaźnik,
                                                                         czyli adres wskaźnika.
int main()
    char tab [] = "321 ffff 1111";
                                                         Funkcja przerywa proces konwersji w
    char *koniecLiczby;
                                                            chwili napotkania znaku, który nie
    long int liczba1, liczba2, liczba3;
                                                                            jest częścią liczby.
   liczba1 = strtol (tab, &koniecLiczby, 10);
   liczba2 = strtol (koniecLiczby, &koniecLiczby, 16);
    liczba3 = strtol (koniecLiczby, &koniecLiczby, 2);
    printf("\nL1: %d, L2: %d, L3: %d\n", liczba1, liczba2, liczba3);
    return 0;
                                         L1: 321, L2: 65535, L3: 15
```



atol Konwertuje wartość zapisaną w łańcuchu znaków do postaci liczby typu całkowitego (long).

Nagłówek funkcji long atol (const char *str) W pliku nagłówkowy stdlib.h zdefiniowane są jeszcze: atoi(), atof().

```
#include <stdio.h>
#include <string.h>
int main() {
    long liczba;
    char tab[] = " 321abc";
    char tab1[] = " a 321abc";

    liczba = atol(tab);
    printf("L1 = %d\n",liczba);

    liczba = atol(tab1);
    printf("L1 = %d\n",liczba);

    return 0;
}
```

- Funkcje te przerywają proces konwersji w momencie napotkania pierwszego znaku, który nie należy do liczby.
- Początkowe znaki białe są pomijane.

```
L1: 321
L1: 0
```



Pliku nagłówkowy stdlib.h

Nagłówek funkcji double strtod (const char *str, char **end)

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    char tab [] = "123.123 69.96";
    char *koniecLiczby;
    double L1, L2;

    L1 = strtod (tab, &koniecLiczby);
    L2 = strtod (koniecLiczby, &koniecLiczby);
    printf("\nL1: %f\n \nL2: %f\n", L1, L2);
    return 0;
}
```

Do obsługi systemu dziesiętnego.

```
L1: 123.123000
L2: 69.960000
```



Biblioteka standardowa języka C

- ➤ **Biblioteka standardowa** języka C dostarcza zestaw funkcji ułatwiających pisanie kodów źródłowych.
- Plik nagłówkowe biblioteki standardowej języka C:

```
<assert.h>
                   <complex.h>
                                       <ctype.h>
<errno.h>
                   \langle fenv.h \rangle
                                       <float.h>
                   <iso646.h>
                                       imits.h>
<inttypes.h>
<locale.h>
                                       <setjmp.h>
                   <math.h>
<signal.h>
                   <stdalign.h>
                                       <stdarg.h>
<stdatomic.h>
                   <stdbool.h>
                                       <stddef.h>
<stdint.h>
                   <stdio.h>
                                       <stdlib.h>
                   <string.h>
                                       <tqmath.h>
<stdnoreturn.h>
                   <time.h>
<threads.h>
                                       <uchar.h>
                   <wctype.h>
<wchar.h>
```

