

21/10/2019

Lista de exercícios  
LP1-ADS

Prof. Luciano Bernardes de Paula

(Lista adaptada do livro “Treinamento em linguagem C” – Victorine Viviane Mizrahi, ed. Pearson).

**Para todos os exercícios, identifique tente sempre usar as funções de manipulação de strings vistas em aula.**

**1 (c7ex29)** Escreva programa que receba uma string e a converta todas as letras para maiúsculas.

Exemplo: a string “abcDEfgh” passa a ser “ABDEFGH”.

Dica: utilize a tabela no final dessa lista.

**2 (c7ex31)** Escreva um programa que receba uma string e um caracter e retorne a primeira posição que esse caracter se encontra na string ou um aviso caso o caracter não existe na string.

Exemplo: caso for recebido a string “programar” e o caracter ‘g’, o programa retorna “posição 3”. Caso for recebido a string “programar” e o caracter ‘x’, o programa retorna “esse caracter não existe na string recebida”.

**3** – Escreva um programa que receba duas strings de mesmo tamanho e retorne se a primeira é maior, menor ou se são iguais na ordem alfabética.

**4** – Escreva um programa que receba duas strings e retorne se a primeira é maior, menor ou se são iguais em relação ao número de caracteres que elas possuem.

**5** Escreva um programa que recebe duas strings str1 e str2 e indica se str2 está contida em str1. Exemplo: a string “baca” está contida em “abacate”.

**6 (c7ex33)** Escreva um programa que receba uma string s, um char c e dois inteiros n e p. O programa preenche n posições da string s com o caracter c a partir da posição p.

Exemplo: se forem recebidas a string “AAAAAAAAAA”, o char c = ‘B’ e os inteiros n = 4 e p = 3, o programa retorna “AAABBBBAAA”

**7 (c7ex34)** Escreva um programa que inverta uma string. Exemplo: abcdef → fedcba

**8 (c7ex35)** Escreva um programa que receba uma string s, um char “atual” e um char “novo”. A função substitui todas as ocorrências do caracter “atual” pelo “novo”.

Exemplo: se for recebida a string “ABCBBDEFBBFF”, o caracter atual B e o caracter novo X, o programa retorna “AXCXXDEFXXFF”.

**9** – Escreva um programa que receba duas strings e concatene-as na primeira.

**10** – Escreva um programa que receba duas strings e copie o conteúdo da primeira na segunda.

**11 (c7ex37)** Escreva um programa que insira uma string em outra a partir da posição indicada (a função recebe string1, string2 e a posição).

Exemplo: string1 recebida “programa”, string2 “yyy” e posição 3. O resultado é “proyyy”.

**12 (c7ex43)** Escreva um programa que receba uma string e retorne a posição do primeiro caractere diferente de um espaço em branco ou um aviso se a string não possuir caracteres ou não possuir caracteres diferentes de um espaço em branco.

**13 (c7ex41)** Escreva um programa que recebe uma string e elimina os caracteres em branco tanto à direita quanto à esquerda.

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	<b>NUL</b> (null)	32	20	040	&#32;	Space	64	40	100	&#64;	@	96	60	140	&#96;	`
1	1	001	<b>SOH</b> (start of heading)	33	21	041	&#33;	!	65	41	101	&#65;	A	97	61	141	&#97;	a
2	2	002	<b>STX</b> (start of text)	34	22	042	&#34;	"	66	42	102	&#66;	B	98	62	142	&#98;	b
3	3	003	<b>ETX</b> (end of text)	35	23	043	&#35;	#	67	43	103	&#67;	C	99	63	143	&#99;	c
4	4	004	<b>EOT</b> (end of transmission)	36	24	044	&#36;	\$	68	44	104	&#68;	D	100	64	144	&#100;	d
5	5	005	<b>ENQ</b> (enquiry)	37	25	045	&#37;	%	69	45	105	&#69;	E	101	65	145	&#101;	e
6	6	006	<b>ACK</b> (acknowledge)	38	26	046	&#38;	&	70	46	106	&#70;	F	102	66	146	&#102;	f
7	7	007	<b>BEL</b> (bell)	39	27	047	&#39;	'	71	47	107	&#71;	G	103	67	147	&#103;	g
8	8	010	<b>BS</b> (backspace)	40	28	050	&#40;	(	72	48	110	&#72;	H	104	68	150	&#104;	h
9	9	011	<b>TAB</b> (horizontal tab)	41	29	051	&#41;	)	73	49	111	&#73;	I	105	69	151	&#105;	i
10	A	012	<b>LF</b> (NL line feed, new line)	42	2A	052	&#42;	*	74	4A	112	&#74;	J	106	6A	152	&#106;	j
11	B	013	<b>VT</b> (vertical tab)	43	2B	053	&#43;	+	75	4B	113	&#75;	K	107	6B	153	&#107;	k
12	C	014	<b>FF</b> (NP form feed, new page)	44	2C	054	&#44;	,	76	4C	114	&#76;	L	108	6C	154	&#108;	l
13	D	015	<b>CR</b> (carriage return)	45	2D	055	&#45;	-	77	4D	115	&#77;	M	109	6D	155	&#109;	m
14	E	016	<b>SO</b> (shift out)	46	2E	056	&#46;	.	78	4E	116	&#78;	N	110	6E	156	&#110;	n
15	F	017	<b>SI</b> (shift in)	47	2F	057	&#47;	/	79	4F	117	&#79;	O	111	6F	157	&#111;	o
16	10	020	<b>DLE</b> (data link escape)	48	30	060	&#48;	0	80	50	120	&#80;	P	112	70	160	&#112;	p
17	11	021	<b>DC1</b> (device control 1)	49	31	061	&#49;	1	81	51	121	&#81;	Q	113	71	161	&#113;	q
18	12	022	<b>DC2</b> (device control 2)	50	32	062	&#50;	2	82	52	122	&#82;	R	114	72	162	&#114;	r
19	13	023	<b>DC3</b> (device control 3)	51	33	063	&#51;	3	83	53	123	&#83;	S	115	73	163	&#115;	s
20	14	024	<b>DC4</b> (device control 4)	52	34	064	&#52;	4	84	54	124	&#84;	T	116	74	164	&#116;	t
21	15	025	<b>NAK</b> (negative acknowledge)	53	35	065	&#53;	5	85	55	125	&#85;	U	117	75	165	&#117;	u
22	16	026	<b>SYN</b> (synchronous idle)	54	36	066	&#54;	6	86	56	126	&#86;	V	118	76	166	&#118;	v
23	17	027	<b>ETB</b> (end of trans. block)	55	37	067	&#55;	7	87	57	127	&#87;	W	119	77	167	&#119;	w
24	18	030	<b>CAN</b> (cancel)	56	38	070	&#56;	8	88	58	130	&#88;	X	120	78	170	&#120;	x
25	19	031	<b>EM</b> (end of medium)	57	39	071	&#57;	9	89	59	131	&#89;	Y	121	79	171	&#121;	y
26	1A	032	<b>SUB</b> (substitute)	58	3A	072	&#58;	:	90	5A	132	&#90;	Z	122	7A	172	&#122;	z
27	1B	033	<b>ESC</b> (escape)	59	3B	073	&#59;	;	91	5B	133	&#91;	[	123	7B	173	&#123;	{
28	1C	034	<b>FS</b> (file separator)	60	3C	074	&#60;	<	92	5C	134	&#92;	\	124	7C	174	&#124;	
29	1D	035	<b>GS</b> (group separator)	61	3D	075	&#61;	=	93	5D	135	&#93;	]	125	7D	175	&#125;	}
30	1E	036	<b>RS</b> (record separator)	62	3E	076	&#62;	>	94	5E	136	&#94;	^	126	7E	176	&#126;	~
31	1F	037	<b>US</b> (unit separator)	63	3F	077	&#63;	?	95	5F	137	&#95;	_	127	7F	177	&#127;	DEL

Source: [www.LookupTables.com](http://www.LookupTables.com)