Tema 9: Cadenas de caracteres.

Operadores y funciones sobre cadenas de caracteres (strcat, strvcat, char, abs, ischar, isnumeric, islogical, num2str, str2num, eval, strcmp, findstr, strfind, length, size, find). Código ascii.

Prof. Saúl. Buitrago y Oswaldo Jiménez

Cadenas de caracteres

A menudo es necesario manipular texto en los programas

- Declaración de un texto

  >> t = ‘esto es un texto’

- Acceso a las entradas de una cadena de caracteres

  >> d = t(1:9) → almacena en la variable d el texto: ‘esto es u’

  de la posición 1 a la 9

  (incluye los espacios en blanco)

  >> d = t(3:10) → almacena en la variable d el texto: ‘to es un’

  de la posición 3 a la 10

Cadena de caracteres

- Operadores sobre cadenas de caracteres

Formas de concatenar texto: strcat y strvcat

>> a = ‘pepe’

Concatenación horizontal:

>> strcat(‘yo me llamo ‘, a, ‘ blanco’) $\rightarrow$ almacena en la variable ans el texto: yo me llamo pepe blanco

Concatenación vertical:

>> strvcat(‘yo me llamo ‘, a, ‘ blanco’) $\rightarrow$ almacena en la variable ans el texto: yo me llamo

pepe

blanco

>> t = [‘yo me llamo’ blanks(1) a blanks(1) ‘blanco’]

Almacena en la variable t el texto: yo me llamo pepe blanco

blanks(1) asigna un espacio en blanco

Cadena de caracteres

- Operadores char y abs

  s = char(x): convierte un entero x (código ascii) a caracter (texto) y lo almacena en la variable s

  x = abs(s): convierte el caracter s (texto) a su código ascii y lo almacena en la variable x

  >> abs(‘a’) → 97

  >> char(97) → a

  >> abs(‘saul’) → 115 97 117 108

  >> char([115 97 117 108]) → saul

  >> char(32) → (espacio en blanco)

  >> abs(‘ ‘) → 32

- Funciones “ischar” y “isnumeric”

  ischar(s) retorna 1 si s es una cadena, en cualquier otro caso 0.

  >> ischar('11') → 1

Ejemplo: uso del “if” múltiple anidado y manejo de cadenas de caracteres

genera el vector “genero” con entradas “f” y “m”.

```
% generacion de los vectores genero, edad e hijos
% estos son almacenados en el archivo 'gen_edad_hijos'
clear
for i=1:1000
    if (mod(i,2)==0), genero(i)='f';
    elseif (mod(i,3)==0), genero(i)='m';
    elseif (mod(i,5)==0), genero(i)='m';
    elseif (mod(i,7)==0), genero(i)='m';
    else
        genero(i)='f';
    end
end
genero = genero';
edad = 14 + round(57*rand(1000,1));
hijos = round(10*rand(1000,1));
save gen_edad_hijos genero edad hijos;
```

archivo : gen_edad_hijos.m

¿Como calcular el número de ‘f’ y ‘m’ en el vector genero?

```
>> vectores_aleatorios
>> load gen_edad_hijos
>> mf = ‘m’-’f’;
cantidad de ‘m’:
>> sum(genero - 'f')/mf
ans = 272
cantidad de ‘f’
>> - sum(genero - 'm')/mf
ans = 728
```

nota: funciones “double”, “char” y “abs”

tabla ascii

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

Cadenas de caracteres

- Operadores num2str y str2num permiten pasar un valor numérico a caracter y viceversa

  >> t = num2str(pi,3) → corresponde al texto: 3.14 (3 dígitos)

  y es almacenado en la variable t

  >> a = [‘el número pi es’ blanks(1) t] → concatena y almacena

  en a el texto: el número pi es 3.14

  >> x = str2num(t) → corresponde al número: 3.14

  >> S = ['1 2'; '3 4']

  >> str2num(S)

```
>> S = ['1 2'; '3 4']
S =
1 2
3 4
>> str2num(S)
ans =
     1     2
     3     4
```

Cadenas de caracteres

- Operador eval permite evaluar una expresión válida en MATLAB pero escrita en forma de texto

  >> x = eval(‘2 + 4’) → interpreta el texto, lo ejecuta como instrucción válida en MATLAB y lo almacena en la variable x

  >> a = eval(‘sin(pi)’) → almacena en a el número 1.2246e-016

  >> for n = 1:4; eval(['M' num2str(n) ' = magic(n) ']); end

  genera una sucesión de matrices con nombres M1 a M4

M1 =

```
     1
```

M2 =

```
     1     3
     4     2
```

M3 =

```
     8     1     6
     3     5     7
     4     9     2
```

M4 =

```
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

Cadenas de caracteres

- Las funciones “length”, “size” y “find” sobre una cadena de caracteres:

  >> a = 'esto es una prueba‘

  >> length(a) retorna el número de caracteres de la cadena, es decir 18

  >> find(a == ‘e’) retorna el vector 1 6 16, correspondientes a las posiciones de la letra “e” sobre el arreglo “a”

  >> S = ['1 2'; '3 4'];

  >> size(S) $\rightarrow$ 2 3

  size(S) retorna el número de filas y columnas de la “matriz”.

  >> find(S == '4') $\rightarrow$ 6

- Función “strcmp” permite comparar cadenas de caracteres, así

  strcmp(s1,s2) retorna 1 si s1 y s2 son iguales, si no retorna 0.

  >> s1 = ‘sss’; s2 = ‘ss’; a = strcmp(s1,s2) $\rightarrow$ a = 0

  >> s1 = ‘sss’; s2 = ‘sss’; a = strcmp(s1,s2) $\rightarrow$ a = 1

Cadena de caracteres

- Función “findstr” permite encontrar una cadena de caracteres sobre otra cadena dada.

  findstr(s1,s2) retorna las posiciones (índice) del primer carácter de la cadena más corta sobre la cadena más larga.

  >> s = ‘Como estas y donde estas?’;

  >> a = findstr(s,’m’) $\rightarrow$ a = 3

  >> a = findstr(s,’estas’) $\rightarrow$ a = [6 20]

  >> a = findstr(s,’Estas’) $\rightarrow$ a = [ ]

  >> a = findstr(s,’ ’) $\rightarrow$ a = [5 11 13 19]

  Obs. Revisar la función “strfind” y comparar con “findstr”.

- Función “inline”: construye una función a partir de una cadena de caracteres,

  Por ejemplo f = inline('cos(x)+2*sin(x) ')

  y así poder calcular f(pi/3) → 2.2321

Ejemplo: construcción de la fórmula de un polinomio dado sus coeficientes en orden decreciente de las potencias como un vector

```
% construucion de la formula de un polinomio dado
% sus coeficientes en orden decreciente de las
% potencias como un vector
p5 = input('ingrese los coeficientes del polinomio  ');
n = length(p5);
pp = [];
for i = 1:n
    ss = '+';
    if sign(p5(i)) == -1
        ss = '-';
    end
    pp = [pp ss num2str(abs(p5(i)),8) '*x.^' num2str(n-i,1)];
end
f = inline(pp);
disp(f);
```

archivo : const_formula_polinomio.m

Cadena de caracteres

Ejercicio:
Dada una cadena de caracteres, la cual representa a un número hexadecimal (es decir, en base 16), hallar el correspondiente valor en base 10.
En este caso es estrictamente necesario manejar el dato de entrada como una cadena de caracteres porque en base 16 se utilizan dígitos representados por letras
(los dígitos en base 16 son 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F).
Se propone que programen una función en Matlab que lleve a cabo esta tarea. Se espera que la función verifique exhaustivamente la pertinencia del parámetro de entrada.
El manejo de los códigos ascii previsto en Matlab facilita enormemente la resolución de este problema.
Se les sugiere resolver primero el problema desde el punto de vista lógico, escribiendo (en el papel) un algoritmo en lenguaje natural, que luego trasladarán al lenguaje de Matlab.

archivo : hexadecimal_a_decimal.m