

Programación 1

**Analista Programador
Analista en Tecnologías de la Información**

Universidad ORT Uruguay

Strings

1 ★☆☆



Solicitar un texto en pantalla y luego mostrarlo en forma invertida.

i Ejemplo, si contiene: 'hola mundo', obtener 'odnum aloh'

2 ★☆☆



Solicitar un texto y una letra e indicar cuántas veces aparece esa letra en el texto Ingresado.

3 ★☆☆



Solicitar un texto e indicar cuantas vocales presenta.

4 ★☆☆



Ingresar un texto y al presionar un botón convertirlo a minúsculas y mostrarlo en un párrafo.

5 ★☆☆



Ingresar un texto y al presionar un botón convertirlo a mayúsculas y mostrarlo en un párrafo.

6 ★☆☆



Ingresar un texto y al presionar un botón convertir todo el texto a minúsculas y solo la primera letra de la primera palabra a mayúsculas.

7 ★☆☆



Ingresa dos textos y verifica si el segundo es una subcadena del primero.

8 ★☆☆



Dado un texto contar la cantidad de palabras que contiene. Para separar palabras en esta etapa consideraremos como separador de palabras únicamente los espacios en blanco.

9 ★☆☆



Dado un texto indicar si comienza y termina con la misma letra.

10 ★☆☆



Sabiendo que la primera letra de una matrícula indica el departamento al que pertenece el vehículo. Ingresa una matrícula (formato: AAA-1111) e informa a que departamento pertenece ese vehículo.

A - Canelones
B - Maldonado
C - Rocha
D - Treinta y Tres
E - Cerro Largo
F - Rivera
G - Artigas
H - Salto
I - Paysandú
J - Río Negro
K - Soriano
L - Colonia
M - San José
N - Flores
O - Florida
P - Lavalleja
Q - Durazno
R - Tacuarembó
S - Montevideo

11 ★★☆☆



Hacer un programa que ingrese una cadena de caracteres y determine el número de mayúsculas y el número de minúsculas.

Se deberán eliminar los espacios en blanco antes de hacer el cálculo y se puede considerar cualquier caracter que no sea a-z o A-Z como mayúscula o minúscula.

En caso de generar una solución más compleja se puede consultar la siguiente tabla.

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

i Con la función `charCodeAt` se puede limitar la verificación a caracteres de la a a la z en mayúscula o minúscula.

12 ★★☆☆



a) Dado un texto Indicar si es un palíndromo. Tener en cuenta que no se tienen en cuenta los espacios en blanco.

Ej: "Amargor pleno con el programa", "Ana lleva al oso la avellana",

b) También se consideran palíndromos los textos en los que no coinciden los acentos o las puntuaciones.

Ej: "Dábele arroz a la zorra el abad", "Saco sal, ayo, del yodo y le doy a las ocas"

i Un palíndromo (del griego *palin dromein*, volver a ir hacia atrás) es una palabra, número o frase que se lee igual hacia adelante que hacia atrás. Si se trata de un número, se llama *capicúa*.

13 ★★☆☆



Dado un texto y una letra mostrar el texto ingresado, remplazando la letra indicada por "*". Por ejemplo si se ingres el texto "tres tristes tigres comen trigo en un trigal" y como letra se ingresa "r", lo que se debe mostrar es: t*es t*istes tig*es comen t*igo en un t*igal

14 ★★☆☆



Crear una función que reciba como parámetro un texto y devuelva la posición en la que se encuentra la penúltima ocurrencia de la letra "a" dentro de ese texto. En caso de que haya una sola "a" o no haya ninguna, devolver -1.

Ej. Para la palabra autoabastecimiento se debe devolver un 4 que es la posición de la penúltima "a".

15 ★★☆☆



Escribir una función que reciba como parámetro una palabra y devuelva la cantidad de veces que aparece la primera letra de esa palabra en toda la palabra.

Ej.: Para la palabra "aumentativo" se debe devolver un 2.

Ej.: Para la palabra "comunicaciones" se debe devolver un 3.

16 ★★★★★



Ingresa una cédula de identidad uruguaya en un campo de texto con formato 1.111.111-1.

Utilizando el algoritmo de cálculo del dígito verificador (el posterior a la barra), verificar si la ingresada es una cédula válida.

Se toman los primeros siete números y se multiplican cada uno (respetando el lugar, es decir: primero x primero, segundo por segundo) por 2987634.

Si la cédula es 1.234.567 la cuenta quedaría así:

$1 \times 2 + 2 \times 9 + 3 \times 8 + 4 \times 7 + 5 \times 6 + 6 \times 3 + 7 \times 4 = 148$.

Se toma el resultado y se le busca el número mayor que termina en 0 y se le resta. Ej: $150 - 148 = 2$.

El número 2 es el dígito verificador para la cédula 1.234.567



Crear un campo de texto para ingresar un valor y verificar si valida con el algoritmo de Luhn.

El **algoritmo de Luhn** o **fórmula de Luhn**, también conocida como "**algoritmo de módulo 10**", es una fórmula de **suma de verificación**, utilizada para validar una diversidad de números de identificación; como **números de tarjetas de crédito**, **números IMEI**, etc.

La fórmula verifica un número contra su dígito de chequeo incluido, el cual es usualmente agregado a un número de cuenta parcial para generar el número de cuenta completo. Este número de cuenta debe pasar la siguiente prueba:

1. A partir del dígito de chequeo incluido, el cual está a la derecha de todo el número, ir de derecha a izquierda duplicando cada segundo dígito.
2. Sumar los dígitos del resultado: (ejemplo: $10 = 1 + 0 = 1$, $14 = 1 + 4 = 5$) juntos con los dígitos sin duplicar del número original.
3. Si el total de **módulo 10** es igual a 0 (si el total termina en cero), entonces el número es válido de acuerdo con la fórmula Luhn, de lo contrario no es válido.

Supongamos un ejemplo de un número de cuenta "7992739871", que contará con un dígito de control adicional, por lo que es de la forma 7992739871x:

Dígitos del número de cuenta	7	9	9	2	7	3	9	8	7	1	x
Duplicar cada siguiente	7	18	9	4	7	6	9	16	7	2	x
Sumar los dígitos	7	9	9	4	7	6	9	7	7	2	=67

El dígito de chequeo (x) se obtiene entonces de $(67 * 9 \text{ mod } 10)$. En términos sencillos:

1. Calcular la suma de los dígitos (67).
2. Multiplicar por 9 (603).
3. Tomar el último dígito (3).
4. El resultado es el dígito de chequeo.