

Manual Tecnico

14/09/2020

Carlos Benjamín Pac Flores Carnet. 201931012
Centro Universitario de Occidente (CUNOC)

Definición de Lenguaje	2
Signos Aceptados Por el lenguaje:	2
Letras:	2
Números:	2
Signos:	2
Palabras Reservadas:	3
Datos Primitivos:	3
Sentencias:	3
Operadores Aritméticos:	3
Operadores Relacionales:	3
Operadores Lógicos:	3
Signos De Agrupación:	4
Finalización de sentencia:	4
Definición de Autómatas	4
ID (referencia de variable):	5
Número Entero:	6
Número Decimal:	7
Cadena:	8
Carácter:	9
Comentario:	10
Diagrama de Clases:	11

Definición de Lenguaje

El lenguaje manejado se basa en palabras reservadas, letras números, signos aritméticos, signos de agrupación, operadores relacionales y operadores lógicos que a continuación se detallan los mismos.

Signos Aceptados Por el lenguaje:

1. Letras:

a	b	c	d	e	f	g	h	i
j	k	l	m	n	ñ	o	p	q
r	s	t	u	v	w	x	y	z

A	B	C	D	E	F	G	H	I
J	K	L	M	N	Ñ	O	P	Q
R	S	T	U	V	W	X	Y	Z

2. Números:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

3. Signos:

	!	"	&	/	()	+	*
-	;	>	<	.				

Palabras Reservadas:

1. Datos Primitivos:

- 1) entero → 34,2
- 2) decimal → 3.4, 678.9, 0.45
- 3) cadena → "esta es una cadena"
- 4) booleano → verdadero,falso
- 5) carácter → s,S,a,A

2. Sentencias:

- 1) SI
- 2) SINO
- 3) SINO_SI
- 4) MIENTRAS
- 5) HACER
- 6) DESDE
- 7) HASTA
- 8) INCREMENTO

Operadores Aritméticos:

- 1) +
- 2) -
- 3) *
- 4) /
- 5) ++
- 6) --
- 7) = (este símbolo también cumple como de asignación de valor)

Operadores Relacionales:

- 1) >
- 2) <
- 3) >=
- 4) <=
- 5) ==
- 6) !=

Operadores Lógicos:

- 1) ||

- 2) &&
- 3) !

Signos De Agrupación:

- 1) (
- 2))

Finalización de sentencia:

- 1) ;

También es de mencionar que se puede admitir el uso de comentarios en el lenguaje con el siguiente formato:

- //Esto es un comentario
- /* Esto también es un comentario */

Definición de Autómatas

Los autómatas son utilizados para el reconocimiento de cadenas así saber al tipo de expresión a la cual pertenece, partiendo desde esta premisa la definición de los autómatas utilizados se basan en expresiones regulares las cuales son:

Referencia	Expresión
ID (referencia a variable)	$[L]^+ [L] \{ [N] \mid [_] \mid ([L]) \}^*$
Número entero	$[0\dots9]^+$
Número decimal	$[0\dots9]^+ \mid [.] [0\dots9]^+$
Cadena	$(\text{"}(\Sigma)^*\text{"})$
Carácter	$([A\dots Z] \mid [a\dots z])$
Comentario	$(/\cdot(\Sigma)^*\cdot(\backslash n)/\cdot'^*\cdot(\Sigma)^*\cdot'^*/)$

En referencia ID se toma "L" como cualquier letra mayúscula o minúscula, "N" como cualquier número y "_" solamente como guión bajo.

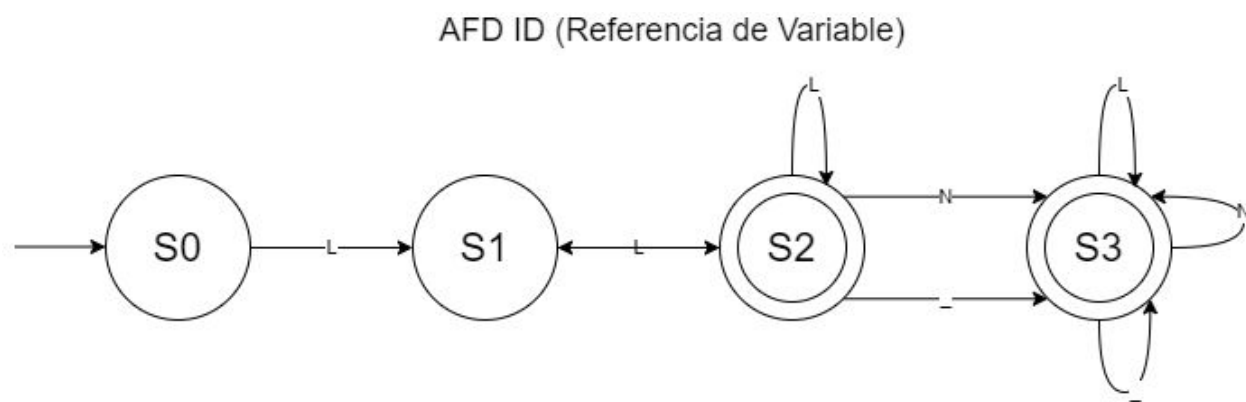
Para los estados booleanos su expresión es considerada como una palabra reservada.

Las palabras reservadas como su nombre lo indica no tienen una expresión regular como tal, sino que son reconocidas automáticamente por jerarquía.

Los autómatas fueron generados por medio del método de Thomson resolviendo las expresiones regulares planteadas anteriormente y algunos de ellos pasaron por una simplificación para llegar al autómata óptimo.

A continuación se presentan los dibujos de los autómatas:

ID (referencia de variable):



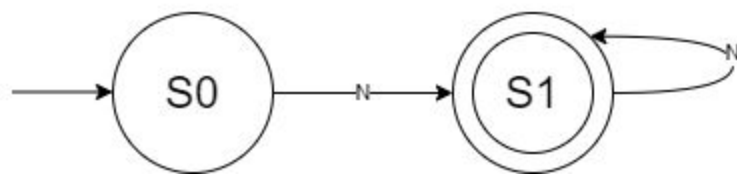
Definición Formal	
Q	S0,S1,S2,S3
Estado Inicial	S0
Alfabeto	{L,N,_} = (letras, números, guión bajo)
Estados De Aceptación	S2,S3

Tabla de Transiciones				
Lenguaje	Estados			
	S0	S1	S2	S3
L	S1	S2	S2	S3

N	\emptyset	\emptyset	S3	S3
-	\emptyset	\emptyset	S3	S3

Número Entero:

ADF Numero Entero

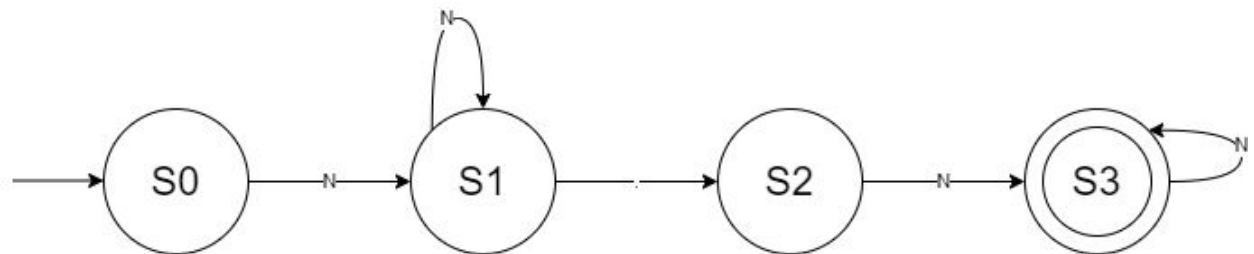


Definición Formal	
Q	S0,S1
Estado Inicial	S0
Alfabeto	{N} (números)
Estados De Aceptación	S1

Tabla de Transiciones		
Lenguaje	Estados	
	S0	S1
N	S1	S1

Número Decimal:

AFD Numero Decimal



Definición Formal

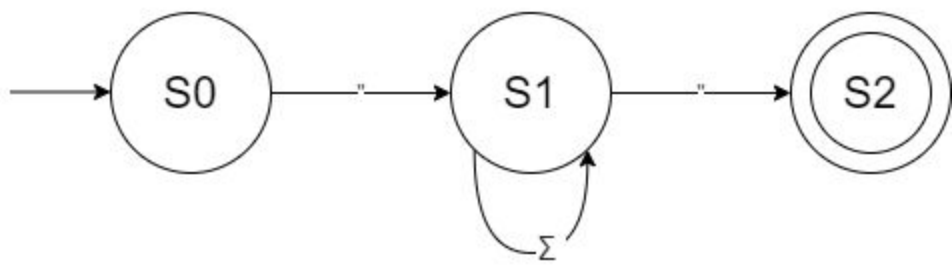
Definición Formal	
Q	S0,S1,S2,S3
Estado Inicial	S0
Alfabeto	{N,"."} (números, punto)
Estados De Aceptación	S3

Tabla de Transiciones

Tabla de Transiciones				
Lenguaje	Estados			
	S0	S1	S2	S3
N	S1	S1	S3	S3
.	∅	S2	∅	∅

Cadena:

AFD Cadena de Texto

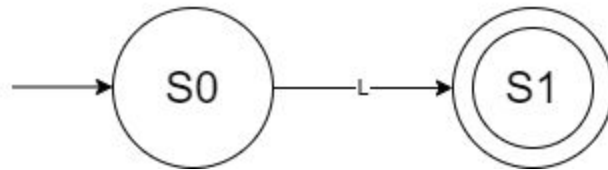


Definición Formal	
Q	S0,S1,S2
Estado Inicial	S0
Alfabeto	{",Σ} (comillas, cualquier carácter)
Estados De Aceptación	S2

Tabla de Transiciones			
Lenguaje	Estados		
	S0	S1	S2
"	S1	S2	∅
Σ	∅	S1	∅

Carácter:

AFD Carcter

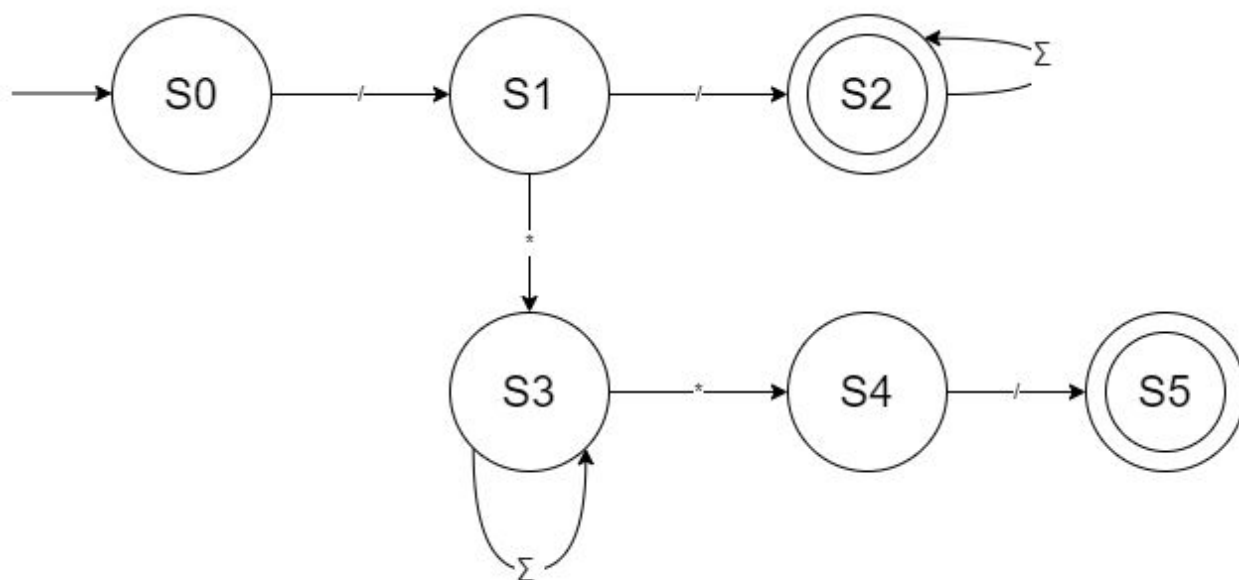


Definición Formal	
Q	S0,S1
Estado Inicial	S0
Alfabeto	{L} (letras)
Estados De Aceptación	S1

Tabla de Transiciones		
Lenguaje	Estados	
	S0	S1
L	S1	∅

Comentario:

AFD Comentario



Definición Formal	
Q	S0,S1,S2,S3,S4,S5
Estado Inicial	S0
Alfabeto	{/,*,Σ} (barra, asterisco, cualquier carácter)
Estados De Aceptación	S2,S5

Tabla de Transiciones						
Lenguaje	Estados					
	S0	S1	S2	S3	S4	S5
Σ	S1	∅	S2	S3	∅	∅
/	∅	S2	∅	∅	S5	∅

--	--	--	--	--	--	--

Se adjunta el diagrama de clases del proyecto adicionalmente en la carpeta contenedora se encontrará el mismo archivo en formato .png para su visualización.

A continuación se presenta la serie de producciones que fueron generadas para el análisis del lenguaje.

$E \rightarrow \text{PRINCIPAL} () \{ \text{EXP} \}$

$\text{EXP} \rightarrow$
ASIG_VAR
| DECLA_VAR
| PRINT
| READ
| ES_SI
| ES_MIENTRAS
| ES_DESDE
| ES_HACER
| ϵ

$\text{PRINT} \rightarrow \text{IMPRIMIR} (\text{OPE});$

$\text{OPE} \rightarrow$
ASIG_E' IDENT
| ASIG_D' IDENT
| ASIG_S' IDENT

$\text{IDENT} \rightarrow$
+ OPE
| ϵ

$\text{READ} \rightarrow \text{LEER} (\text{ID_CADENA});$

$\text{ASIG_VAR} \rightarrow$
ID_ENTERO ASIG_VAR_E
| ID_DECIMAL ASIG_VAR_D
| ID_BOOLEANO = ASIG_B';
| ID_CARACTER = ASIG_C';
| ID_CADENA = ASIG_S';

$\text{ASIG_VAR_E} \rightarrow$
++;
| --;
| = ASIG_E';

$\text{ASIG_VAR_D} \rightarrow ++;$

| --;
| = ASIG_D' ;

DECLA_VAR → ENTERO ASIG_E
| BOOLEANO ASIG_B
| DECIMAL ASIG_D
| CADENA ASIG_S
| CARACTER ASIG_C

ASIG_E → = ASIG_E' ;
| , ID_ENTERO ASIG_E
| ID_ENTERO ASIG_E
| ;

ASIG_E' → - ASIG_E"
| ASIG_E"

ASIG_E" → NUMERO_E
| ID_ENTERO

ASIG_D → = ASIG_D' ;
| , ID_DECIMAL ASIG_D
| ID_DECIMAL ASIG_D
| ;

ASIG_D' → - ASIG_D"
| ASIG_D"

ASIG_D" → NUMERO_D
| ID_DECIMAL
| ID_ENTERO
| NUMERO_E

ASIG_B → = ASIG_B' ;

| , ID_BOOLEANO ASIG_B
 | ID_BOOLEANO ASIG_B
 | ;

ASIG_B' → VERDADERO
 | FALSO
 | ID_BOOLEANO

ASIG_S → = ASIG_S' ;
 | , ID_CADENA ASIG_S
 | ID_CADENA ASIG_S
 | ;

ASIG_S' → ID_CADENA
 | CAD_TEXTO

ASIG_C → = ASIG_C' ;
 | , ID_CARACTER ASIG_C
 | ID_CARACTER ASIG_C
 | ;

ASIG_C' → LETRA
 | ID_CARACTER

ES_MIENTRAS → MIENTRAS (LOGICA) { EXP }

ES_SI → SI (LOGICA) { EXP } ES_SI'

ES_SI' → SINO { EXP }
 | SINO_SI (LOGICA) { EXP } ES_SI'
 | ε

ES_HACER → HACER { EXP } MIENTRAS (LOGICA)

ES_DESDE → DESDE ASIG_DES HASTA LOGICA INCREMENTO ASIG_DES" { EXP }

ASIG_DES → ID_ENTERO ASIG_DES'
| ENTERO ID_ENTERO ASIG_DES'

ASIG_DES' → = ASIG_DES"

ASIG_DES" → ID_ENTERO
| NUMERO_E

LOGICA → ASIG_D" OP_RELACIONAL ASIG_D" LOGICA'
| ASIG_B' LOGICA'
| ! LOGICA LOGICA'

LOGICA' → && LOGICA
| || LOGICA
| ε

Cálculo de Primeros

E	PRINCIPAL
EXP	ID_ENTERO, ID_DECIMAL, ID_BOOLEANO, ID_CARACTER, ENTERO, BOOLEANO, DECIMAL, CADENA, CARÁCTER, IMPRIMIR, LEER, SI, MIENTRAS, DESDE, HACER, e
PRINT	IMPRIMIR
OPE	-, NUMERO_E, ID_ENTERO, ID_DECIMAL, NUMERO_D, ID_CADENA, CAD_TEXTO
IDENT	+, e
READ	LEER
ASIG_VAR	ID_ENTERO, ID_DECIMAL, ID_BOOLEANO, ID_CARACTER, ID_CADENA
ASIG_VAR_E	++, --, =

ASIG_VAR_D	++, --, =
DECLA_VAR	ENTERO, BOOLEANO, DECIMAL, CADENA, CARACTER
ASIG_E	=, ,, ID_ENTERO, ;
ASIG_E'	-, NUMERO_E, ID_ENTERO
ASIG_E''	NUMERO_E, ID_ENTERO
ASIG_D	=, ,, ID_DECIMAL, ;
ASIG_D'	-, NUMERO_D, NUMERO_E, ID_ENTERO, ID_DECIMAL
ASIG_D''	NUMERO_D, NUMERO_E, ID_ENTERO, ID_DECIMAL
ASIG_B	=, ,, ID_BOOLEANO, ;
ASIG_B'	VERDADERO, FALSO, ID_BOOLEANO
ASIG_S	=, ,, ID_CADENA, ;
ASIG_S'	ID_CADENA, CAD_TEXTO
ASIG_C	=, ,, ID_CARACTER, ;
ASIG_C'	CARACTER, ID_CARACTER
ES_MIENTRAS	MIENTRAS
ES_SI	SI
ES_SI'	SINO, SINO_SI, e
ES_HACER	HACER
ES_DESDE	DESDE
ASIG_DES	ID_ENTERO, ENTERO
ASIG_DES'	=
ASIG_DES''	ID_ENTERO, NUMERO_E
LOGICA	NUMERO_D, NUMERO_E, ID_ENTERO, ID_DECIMAL, VERDADERO, FALSO, ID_BOOLEANO, !
LOGICA'	&&, , e

Cálculo de Siguientes:

E	\$
EXP	}
PRINT	ID_ENTERO, ID_DECIMAL, ID_BOOLEANO, ID_CHARACTER, ENTERO, BOOLEANO, DECIMAL, CADENA, CARÁCTER, IMPRIMIR, LEER, SI, MIENTRAS, DESDE, HACER, e
OPE),+,e
IDENT	-, NUMERO_E, ID_ENTERO, NUMERO_D, ID_DECIMAL, ID_CADENA, CAD_TEXTO
READ	ID_ENTERO, ID_DECIMAL, ID_BOOLEANO, ID_CHARACTER, ENTERO, BOOLEANO, DECIMAL, CADENA, CARÁCTER, IMPRIMIR, LEER, SI, MIENTRAS, DESDE, HACER, e
ASIG_VAR	ID_ENTERO, ID_DECIMAL, ID_BOOLEANO, ID_CHARACTER, ENTERO, BOOLEANO, DECIMAL, CADENA, CARÁCTER, IMPRIMIR, LEER, SI, MIENTRAS, DESDE, HACER, e
ASIG_VAR_E	ID_ENTERO, ID_DECIMAL, ID_BOOLEANO, ID_CHARACTER, ID_CADENA
ASIG_VAR_D	ID_ENTERO, ID_DECIMAL, ID_BOOLEANO, ID_CHARACTER, ID_CADENA
DECLA_VAR	ID_ENTERO, ID_DECIMAL, ID_BOOLEANO, ID_CHARACTER, ENTERO, BOOLEANO, DECIMAL, CADENA, CARÁCTER, IMPRIMIR, LEER, SI, MIENTRAS, DESDE, HACER, e
ASIG_E	ENTERO, BOOLEANO, DECIMAL, CADENA, CARÁCTER, , , ID_ENTERO, ; , =
ASIG_E'	++, --, =, , , , ; , -, NUMERO_E, ID_ENTERO, ID_DECIMAL, NUMERO_D, ID_CADENA, CAD_TEXTO
ASIG_E''	-, NUMERO_E, ID_ENTERO
ASIG_D	=, , , ID_DECIMAL, ; , ENTERO, BOOLEANO, DECIMAL, CADENA, CHARACTER
ASIG_D'	++, --, =, , , , ; , -, NUMERO_E, ID_ENTERO, ID_DECIMAL, NUMERO_D, ID_CADENA, CAD_TEXTO

ASIG_D"	-, NUMERO_D, NUMERO_E, ID_ENTERO , ID_DECIMAL, NUMERO_D, NUMERO_E , VERDADERO, FALSO, ID_BOOLEANO, !, && , , e ,) , INCREMENTO
ASIG_B	ENTERO, BOOLEANO, DECIMAL, CADENA, CARÁCTER, = , , , ID_BOOLEANO, ;
ASIG_B'	; , = , , , ID_BOOLEANO , && , ,) , INCREMENTO , e
ASIG_S	= , , , ID_CADENA, ; , ENTERO, BOOLEANO, DECIMAL, CADENA, CARACTER
ASIG_S'	; , = , , , ID_CADENA, -, NUMERO_E, ID_ENTERO, ID_DECIMAL, NUMERO_D, ID_CADENA, CAD_TEXTO
ASIG_C	ENTERO, BOOLEANO, DECIMAL, CADENA, CARÁCTER , = , , , ID_CARACTER, ;
ASIG_C'	; , = , ID_CARACTER
ES_MIENTRAS	ID_ENTERO, ID_DECIMAL, ID_BOOLEANO, ID_CARACTER, ENTERO, BOOLEANO, DECIMAL, CADENA, CARÁCTER, IMPRIMIR, LEER, SI, MIENTRAS, DESDE, HACER, e
ES_SI	ID_ENTERO, ID_DECIMAL, ID_BOOLEANO, ID_CARACTER, ENTERO, BOOLEANO, DECIMAL, CADENA, CARÁCTER, IMPRIMIR, LEER, SI, MIENTRAS, DESDE, HACER, e
ES_SI'	SI, SINO, SINO_SI , e
ES_HACER	ID_ENTERO, ID_DECIMAL, ID_BOOLEANO, ID_CARACTER, ENTERO, BOOLEANO, DECIMAL, CADENA, CARÁCTER, IMPRIMIR, LEER, SI, MIENTRAS, DESDE, HACER, e
ES_DESDE	ID_ENTERO, ID_DECIMAL, ID_BOOLEANO, ID_CARACTER, ENTERO, BOOLEANO, DECIMAL, CADENA, CARÁCTER, IMPRIMIR, LEER, SI, MIENTRAS, DESDE, HACER, e
ASIG_DES	HASTA
ASIG_DES'	ID_ENTERO , ENTERO,
ASIG_DES"	= , {
LOGICA	&& , , e ,) , INCREMENTO
LOGICA'	NUMERO_D, NUMERO_E, ID_ENTERO, ID_DECIMAL, VERDADERO, FALSO, ID_BOOLEANO, !

Tabla de Transiciones:

Debido al tamaño de la tabla de transiciones esta se adjunta, en formato PNG para mejor visualización:

3000																															
E	PROCESO (11000)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
EXP						PROG																									
PROG						PROG (EXP)																									
EXP																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															
PROG																															

USO DE TRANSICIONES																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
---------------------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--