

## ALFA Programming Language Grammar

1	<programa>	::=	<b>main</b> { <declaraciones> <funciones> <sentencias> }
2	<declaraciones>	::=	<declaracion>
3			<declaracion> <declaraciones>
4	<declaracion>	::=	<clase> <identificadores> ;
5	<clase>	::=	<clase_escalor>
6			<clase_puntero>
7			<clase_vector>
8			<clase_conjunto>
9	<clase_escalor>	::=	<tipo>
10	<tipo>	::=	<b>int</b>
11			<b>boolean</b>
12			<b>float</b>
13	<clase_puntero>	::=	<tipo> *
14			<clase_puntero> *
15	<clase_vector>	::=	<b>array</b> <tipo> [ <constante_entera> ]
16			<b>array</b> <tipo> [ <constante_entera> , <constante_entera> ]
17	<clase_conjunto>	::=	<b>set of</b> <constante_entera>
18	<identificadores>	::=	<identificador>
19			<identificador> , <identificadores>
20	<funciones>	::=	<funcion> <funciones>
21			
22	<funcion>	::=	<b>function</b> <tipo> <identificador> ( <parametros_funcion> ) {
23	<parametros_funcion>	::=	<parametro_funcion> <resto_parametros_funcion>
24			
25	<resto_parametros_funcion>	::=	; <parametro_funcion> <resto_parametros_funcion>
26			
27	<parametro_funcion>	::=	<tipo> <identificador>
28	<declaraciones_funcion>	::=	<declaraciones>
29			
30	<sentencias>	::=	<sentencia>
31			<sentencia> <sentencias>
32	<sentencia>	::=	<sentencia_simple> ;
33			<bloque>
34	<sentencia_simple>	::=	<asignacion>
35			<lectura>
36			<escritura>
37			<liberacion>
38			<retorno_funcion>
39			<operacion_conjunto>
40	<bloque>	::=	<condicional>
41			<bucle>
42			<seleccion>
43	<asignacion>	::=	<identificador> = <exp>
44			<elemento_vector> = <exp>
45			<acceso> = <exp>
46			<identificador> = <b>malloc</b>
47			<identificador> = <b>&amp;</b> <identificador>
48	<elemento_vector>	::=	<identificador> [ <exp> ]
49			<identificador> [ <exp> , <exp> ]
50	<condicional>	::=	<b>if</b> ( <exp> ) { <sentencias> }
51			<b>if</b> ( <exp> ) { <sentencias> } <b>else</b> { <sentencias> }
52	<bucle>	::=	<b>while</b> ( <exp> ) { <sentencias> }
53			<b>for</b> ( <identificador> = <exp> ; <exp> ) { <sentencias> }
54	<lectura>	::=	<b>scanf</b> <identificador>
55			<b>scanf</b> <elemento_vector>
56	<escritura>	::=	<b>printf</b> <exp>
57			<b>cprintf</b> <identificador>
58	<liberacion>	::=	<b>free</b> <identificador>

59	<acceso>	::=	* <identificador>
60			* <acceso>
61	<retorno_funcion>	::=	<b>return</b> <exp>
62	<seleccion>	::=	<b>switch</b> ( <exp> ) { <casos_seleccion> }
63	<casos_seleccion>	::=	<casos_estandar> <caso_defecto>
64	<casos_estandar>	::=	<caso_estandar>
65			<casos_estandar> <caso_estandar>
66	<caso_estandar>	::=	<b>case</b> <constante_entera> : <sentencias>
67	<caso_defecto>	::=	<b>default</b> <sentencias>
68	<operacion_conjunto>	::=	<b>union</b> ( <identificador> , <identificador> , <identificador> )
69			<b>intersection</b> ( <identificador> , <identificador> , <identificador> )
70			<b>add</b> ( <exp> , <identificador> )
71			<b>clear</b> ( <identificador> )
72	<exp>	::=	<exp> + <exp>
73			<exp> - <exp>
74			<exp> / <exp>
75			<exp> * <exp>
76			- <exp>
77			<exp> && <exp>
78			<exp>    <exp>
79			! <exp>
80			<identificador>
81			<constante>
82			( <exp> )
83			( <comparacion> )
84			<acceso>
85			<elemento_vector>
86			<b>size</b> ( <identificador> )
87			<b>contains</b> ( <exp> , <identificador> )
88			<identificador> ( <lista_expresiones> )
89	<lista_expresiones>	::=	<exp> <resto_lista_expresiones>
90			
91	<resto_lista_expresiones>	::=	, <exp> <resto_lista_expresiones>
92			
93	<comparacion>	::=	<exp> == <exp>
94			<exp> != <exp>
95			<exp> <= <exp>
96			<exp> >= <exp>
97			<exp> < <exp>
98			<exp> > <exp>
99	<constante>	::=	<constante_logica>
100			<constante_entera>
101			<constante_real>
102	<constante_logica>	::=	<b>true</b>
103			<b>false</b>
104	<constante_entera>	::=	<numero>
105	<numero>	::=	<digito>
106			<numero> <digito>
107	<constante_real>	::=	<constante_entera> . <constante_entera>
108	<identificador>	::=	<letra>
109			<letra> <cola_identificador>
110	<cola_identificador>	::=	<alfanumerico>
111			<alfanumerico> <cola_identificador>
112	<alfanumerico>	::=	<letra>
113			<digito>
114	<letra>	::=	<b>a   b   ...   z   A   B   ...   Z</b>
115	<digito>	::=	<b>0   1   2   3   4   5   6   7   8   9</b>

### **Additional considerations**

- The language allows to include comments between the characters of // and the end of line (they are one-line comments).
- Identifiers length is limited to 100 characters.

**Only the productions highlighted in grey are going to be used in the compiler for this course**