

## Convenciones utilizadas en la redacción de problemas que afectan a los prototipos Pascal

Forma de citarlo en los enunciados de los problemas	Parámetro de entrada	Parámetro de salida	Resultado de función
(nn: ) ENTERO	nn : <b>LongInt</b>	<b>out</b> nn : <b>LongInt</b>	: <b>LongInt</b>
(nn: ) ENTERO LARGO	nn : <b>Int64</b>	<b>out</b> nn : <b>Int64</b>	: <b>Int64</b>
(nn: ) CARACTER	nn : <b>Char</b>	<b>out</b> nn : <b>Char</b>	: <b>Char</b>
(nn: ) CADENA (*)	nn : <b>ansistring</b>	<b>out</b> nn : <b>ansistring</b>	: <b>ansistring</b>
(nn: ) PALABRAS (+)	nn : <b>ansistring</b>	<b>out</b> nn : <b>ansistring</b>	: <b>ansistring</b>
(nn: ) ARREGLO[ N ] de T (×)	nn : <b>var array of T</b>	<b>out</b> nn : <b>array of T</b>	: <b>array of T</b>
(nn: ) MATRIZ[ M, N ] de T (×)	Las matrices se obtienen con arreglos cuyo T es a su vez un arreglo		
(nn: ) REGISTRO[ s <sub>1</sub> s <sub>2</sub> T,... ]	nn : <b>record</b> s <sub>1</sub> , s <sub>2</sub> : T; ...end		

(\*) Se sobrentiende que sólo puede haber caracteres ASCII en el rango [32,126] y terminan con ASCII 0.

(+) Se sobrentiende que sólo puede haber espacios y letras mayúsculas y terminan con ASCII 0.

(×) T designa un tipo, N es una variable del enunciado. Lo mismo M.

En las entradas el evaluador local habrá adecuado el `length` a N ( y M )

En las salidas y retornos de función lo deberá hacer el programador con `setLength`

Las referencias en el texto a primer, segundo o i-ésimo elemento se refieren a su ubicación física sin tener en cuenta la forma de referenciarlo en algún lenguaje en particular.

## Convenciones utilizadas en la redacción de problemas que afectan a los prototipos C++

Forma de citarlo en los enunciados de los problemas	Parámetro de entrada	Parámetro de salida	Resultado de función
(nn:) ENTERO	<b>int</b> nn	<b>int &amp;</b> nn	<b>int</b> f()
(nn:) ENTERO LARGO	<b>long long</b> nn	<b>long long &amp;</b> nn	<b>long long</b> f()
(nn:) CARACTER	<b>char</b> nn	<b>char &amp;</b> nn	<b>char</b> f()
(nn:) CADENA (*)	<b>string</b> nn	<b>string &amp;</b> nn	<b>string</b> f()
(nn:) PALABRAS (+)	<b>string</b> nn	<b>string &amp;</b> nn	<b>string</b> f()
(nn:) ARREGLO[ N ] de T (x)	<b>vector&lt; T &gt;</b> nn	<b>vector&lt; T &gt; &amp;</b> nn	<b>vector&lt; T &gt;</b> f()
(nn:) MATRIZ[ M, N ] de T (x)	Las matrices se obtienen con arreglos cuyo T es a su vez un arreglo		
(nn:) REGISTRO[ s <sub>1</sub> s <sub>2</sub> T,... ]	<b>struct{ T s<sub>1</sub> ; T s<sub>2</sub> ;...}</b> nn		

(\*) Se sobrentiende que sólo puede haber caracteres ASCII en el rango [32,126] y terminan con ASCII 0.

(+) Se sobrentiende que sólo puede haber espacios y letras mayúsculas y terminan con ASCII 0.

(x) T designa un tipo, N es una variable del enunciado . Lo mismo M.

En las entradas el evaluador habrá adecuado el `.size()` a N ( y M )

En las salidas y retornos de función lo deberá hacer el programador con `.resize()`

Las referencias en el texto a primer, segundo o i-ésimo elemento se refieren a su ubicación física sin tener en cuenta la forma de referenciarlo en algún lenguaje en particular.