

Summary on records

	Algorithm	C language (method 1) Definition by <i>typedef</i>	C language (method 2) Definition by <i>struct</i>
Type definition	<p>Type Record_Name=Record Field1: Type1; Field2: Type2; Field_n:Type_n; END</p> <p>Example : Type Date = Record day: integer; month:integer; year: integer; END</p>	<pre>typedef struct Record_name{ Type1 Field1; Type2 Field2; Type_n Field_n; } Record_name;</pre> <p>Example : typedef struct Date {int day ; int day; int month ; int year; } Date;</p>	<pre>struct Record_Name { Type1 Field1; ... Type_n Field_n; };</pre> <p>Example : struct Date { int day; int month; int year; };</p>
Declaration of variables	<p>Variable_Name: Record_Name; Example: d: Date;</p>	<p>Record_Name Variable_Name ; Example: Date d;</p>	<p>struct Record_Name Variable_Name; Example: struct Date d;</p>
Access to variable fields	<p>Variable_Name. Field_Name; Example: d.day=3; Read (d.month);</p>	<p>Variable_Name.Field_Name; Example : d.day=3; scanf ("% d", d.month);</p>	<p>Variable_Name.Field_Name; Example : d.day=3; scanf ("% d", d.month);</p>
Record_Array	<p>array_name: array [1..N] of Record_Name; Example: T: array [1..50] of Date;</p>	<p>Record_name array_name [N]; Example : Date T[50];</p>	<p>struct Record_name array_name [N]; Example : struct Date T[50];</p>