

## OII - Pseudocodice(4.0)

Pseudocodice	Descrizione	Esempio
<b>■ Variabili e tipi</b>		
<b>variable</b> <i>i</i> : <i>integer</i>	Dichiarazione della variabile di tipo intero chiamata <b>i</b>	
<b>variable</b> <i>arr</i> : <i>integer</i> []	Dichiarazione di una variabile di tipo array di interi chiamata <b>arr</b>	
{var} ← {espr}	Assegnamento del valore dell'espressione {espr} alla variabile {var}	<b>i</b> ← 1 <b>a</b> ← 3 × <b>i</b> + 5 <b>arr</b> ← [3, 5/a, 2]
<b>arr</b> [{espr}]	Variabile corrispondente all'elemento dell'array <b>arr</b> di indice {espr}	<b>a</b> ← <b>arr</b> [3] + 1 <b>arr</b> [ <b>x</b> + 1] ← 2
( <b>a</b> , <b>b</b> ) ← ( <b>b</b> , <b>a</b> )	Scambio del valore delle variabili <b>a</b> e <b>b</b>	
<b>■ Operatori</b>		
+, −, ×, /, mod	Aritmetica: addizione, sottrazione (o negazione), moltiplicazione, divisione intera, resto della divisione intera (modulo)	<b>a</b> + <b>b</b> − <b>a</b> <b>i</b> ← <b>a</b> mod 10
==, ≠, <, ≤, >, ≥	Confronto: uguale, diverso, minore, minore o uguale, maggiore, maggiore o uguale	<b>a</b> == 7 2 × ( <b>x</b> + 1) ≤ <b>y</b>
and, or, not	Operatori logici: e, o, non	<b>a</b> > <b>b</b> and <b>b</b> ≠ -1 not ( <b>a</b> > 2 or <b>a</b> == 0)
<b>■ Strutture di controllo</b>		
<b>if</b> {condizione} <b>then</b> {corpo if} <b>else</b> {corpo else} <b>end if</b>	Struttura condizionale <b>if ... else</b> : se {condizione} è vera viene eseguito {corpo if}, altrimenti viene eseguito {corpo else}. La parte <b>else</b> può essere omessa	<b>if</b> <b>n</b> mod 2 == 0 <b>then</b> <b>i</b> ← 0 <b>else</b> <b>i</b> ← <b>n</b> − 1 <b>end if</b>
<b>while</b> {condizione} <b>do</b> {corpo} <b>end while</b>	Ciclo <b>while</b> : il blocco {corpo} viene ripetuto fintanto che {condizione} è vera	<b>while</b> <b>i</b> < <b>n</b> <b>do</b> <b>sum</b> ← <b>sum</b> + <b>i</b> <b>i</b> ← <b>i</b> + 7 <b>end while</b>
<b>for</b> {indice} <b>in</b> {intervallo} <b>do</b> {corpo} <b>end for</b>	Ciclo <b>for</b> : il blocco {corpo} viene eseguito mentre la variabile {indice} itera sui valori in {intervallo}, specificato come [ <i>a</i> ... <i>b</i> ], che significa "tutti i numeri da <i>a</i> (incluso) fino a <i>b</i> (escluso)"	<b>for</b> <b>i</b> <b>in</b> [0 ... <b>n</b> ] <b>do</b> <b>arr</b> [ <b>i</b> ] ← -1 <b>end for</b> (assegna -1 a tutti gli elementi di un array <b>arr</b> di lunghezza <b>n</b> )
<b>■ Funzioni</b>		
<b>function</b> <b>fun</b> ( <b>var1</b> : <i>tipo1</i> , <b>var2</b> : <i>tipo2</i> , ...) → <i>ritorno</i> {corpo} <b>end function</b>	Funzione con parametri <b>var1</b> , <b>var2</b> , etc. Il tipo di ritorno → <i>ritorno</i> può essere omesso. Il valore viene restituito tramite la parola chiave <b>return</b>	<b>function</b> add( <b>a</b> : <i>integer</i> , <b>b</b> : <i>integer</i> ) → <i>integer</i> <b>return</b> <b>a</b> + <b>b</b> <b>end function</b>
<b>fun</b> () <b>fun</b> ( <b>arg1</b> , <b>arg2</b> , ...)	Chiamata alla funzione <b>fun</b> (rispettivamente senza argomenti e con argomenti). La funzione <b>output</b> stampa il valore di una variabile oppure una stringa fissata. Le funzioni <b>min</b> e <b>max</b> restituiscono risp. il minimo e il massimo di due interi	<b>return</b> add( <b>a</b> , <b>b</b> ) <b>m</b> ← very_big_integer() <b>output</b> ( <b>x</b> ) <b>output</b> ("string") <b>min</b> ( <b>a</b> , <b>b</b> ) <b>max</b> ( <b>a</b> , <b>b</b> )