

Pseudocodice	Descrizione	Esempio
■ Variabili e tipi		
<b>variable</b> <i>i</i> : <i>integer</i>	Dichiarazione della variabile di tipo intero chiamata <i>i</i>	
<b>variable</b> <i>arr</i> : <i>integer[]</i>	Dichiarazione di una variabile di tipo array di interi chiamata <i>arr</i>	
<b>{var}</b> ← <b>{espr}</b>	Assegnamento del valore dell'espressione <b>{espr}</b> alla variabile <b>{var}</b>	<i>i</i> ← 1 <i>a</i> ← 3 × <i>i</i> + 5 <i>arr</i> ← [3, 5/ <i>a</i> , 2]
<i>arr</i> [ <b>{espr}</b> ]	Variabile corrispondente all'elemento dell'array <i>arr</i> di indice <b>{espr}</b>	<i>a</i> ← <i>arr</i> [3] + 1 <i>arr</i> [ <i>x</i> + 1] ← 2
( <i>a</i> , <i>b</i> ) ← ( <i>b</i> , <i>a</i> )	Scambio del valore delle variabili <i>a</i> e <i>b</i>	
■ Operatori		
+, −, ×, /, <b>mod</b>	Aritmetica: addizione, sottrazione (o negazione), moltiplicazione, divisione intera, resto della divisione intera (modulo)	<i>a</i> + <i>b</i> − <i>a</i> <i>i</i> ← <i>a mod</i> 10
==, ≠, <, ≤, >, ≥	Confronto: uguale, diverso, minore, minore o uguale, maggiore, maggiore o uguale	<i>a</i> == 7 2 × ( <i>x</i> + 1) ≤ <i>y</i>
<b>and</b> , <b>or</b> , <b>not</b>	Operatori logici: e, o, non	<i>a</i> > <i>b</i> <b>and</b> <i>b</i> ≠ −1 <b>not</b> ( <i>a</i> > 2 <b>or</b> <i>a</i> == 0)
■ Strutture di controllo		
<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> <i>n mod</i> 2 == 0 <b>then</b> <i>i</i> ← 0 <b>else</b> <i>i</i> ← <i>n</i> − 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> <i>i</i> < <i>n</i> <b>do</b> <i>sum</i> ← <i>sum</i> + <i>i</i> <i>i</i> ← <i>i</i> + 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> <i>i</i> <b>in</b> [0 ... <i>n</i> ] <b>do</b> <i>arr</i> [ <i>i</i> ] ← −1 <b>end for</b>  (assegna −1 a tutti gli elementi di un array <i>arr</i> di lunghezza <i>n</i> )
■ Funzioni		
<b>function</b> <b>fun</b> ( <i>var1</i> : <i>tipo1</i> , <i>var2</i> : <i>tipo2</i> , ...) → <i>ritorno</i> {corpo} <b>end function</b>	Funzione con parametri <i>var1</i> , <i>var2</i> , 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> <b>add</b> ( <i>a</i> : <i>integer</i> , <i>b</i> : <i>integer</i> ) → <i>integer</i> <b>return</b> <i>a</i> + <i>b</i> <b>end function</b>
<b>fun</b> () <b>fun</b> ( <i>arg1</i> , <i>arg2</i> , ...)	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> <b>add</b> ( <i>a</i> , <i>b</i> ) <i>m</i> ← <b>very_big_integer</b> () <b>output</b> ( <i>x</i> ) <b>output</b> ("string") <b>min</b> ( <i>a</i> , <i>b</i> ) <b>max</b> ( <i>a</i> , <i>b</i> )