

## OII - Pseudocodice(3.0)

Pseudocodice	Significato
<b>variable</b> a: integer <b>variable</b> w: integer[14]  <b>variable</b> mat: char[10][10]	dichiarazione della variabile a di tipo intero dichiarazione della variabile w di tipo array di interi fatto da 14 elementi, indicizzati (come nel linguaggio C) da w[0] a w[13] dichiarazione della variabile mat di tipo matrice di caratteri di dimensione 10 × 10, indicizzati (come nel linguaggio C) da mat[0][0] a mat[9][9]
a ← 3 a ← a + 1	assegnamento del valore 3 ad a, incremento del valore di a
<b>output</b> a	istruzione per scrivere dati sullo schermo
<b>output</b> "ciao" <b>output</b> "ciao", var	scrive la stringa ciao scrive ciao e poi il contenuto della variabile var
array[7] ← 0 mat[5][28] ← 1	esempi di inizializzazione di un elemento di un array e di una cella di una matrice
+ - × / MOD	questi simboli indicano rispettivamente le operazioni aritmetiche di somma, differenza, prodotto, divisione e resto della divisione intera
= < > ≤ ≥ ≠	sono i simboli che indicano i sei tipi di confronto: uguale, minore, maggiore, minore o uguale, maggiore o uguale, diverso
(a ≤ 3) <b>or</b> (a ≥ 5) (a ≤ 3) <b>and</b> (a ≤ 5) <b>not</b> (a)	esempi di operazioni logiche: <b>not</b> per il NOT logico, <b>and</b> per l'AND, <b>or</b> per l'OR

**Attenzione!** Queste definizioni non sono un "contratto vincolante". Il Comitato Olimpico potrebbe ritenere utile usare notazioni speciali ai fini della chiarezza e facilità di lettura dell'esercizio. Per esempio:

(a ≤ 3) **or** (b è un numero primo)

Esempi di strutture di controllo (equivalenti di if, while, etc ...)

Pseudocodice	Significato
<b>if</b> condizione <b>then</b> ... <b>end if</b>	è un costrutto condizionale
<b>while</b> condizione <b>do</b> ... <b>end while</b>	è un ciclo a condizione iniziale (pre-posta), ovvero il corpo viene eseguito ogni volta che <b>condizione</b> è VERA
<b>do</b> ... <b>while</b> condizione	è un ciclo a condizione finale (post-posta), ovvero il corpo viene eseguito fino a che <b>condizione</b> diventa FALSA
<b>function</b> CALCOLA(p1: tipo, ..., pn: tipo) → tipo ... <b>return</b> valore_restituito <b>end function</b>	sintassi di una funzione che accetta dei parametri (ciascuno con un tipo) e restituisce un certo valore
<b>procedure</b> ESEGUI(p1: tipo, ..., pn: tipo) ... <b>return</b> <b>end procedure</b>	sintassi di una procedura (funzione che <b>non</b> restituisce un valore)