

Nome costruito	In linguaggio C	Pseudocodifica	Flow-chart
Sequenza	<pre> <istruzione_1>; <istruzione_2>; <istruzione_3>; </pre>	<pre> <istruzione_1> <istruzione_2> <istruzione_3> </pre>	
Selezione binaria (chiamata anche scelta binaria)	<pre> if (<condizione>) { <istruzione_1>; } else { <istruzione_2>; } </pre>	<pre> SE (<condizione>) ALLORA <istruzione_1> ALTRIMENTI <istruzione_2> FINE SE </pre>	
Selezione unaria (chiamata anche scelta unaria)	<pre> if (<condizione>) { <istruzione_1>; } </pre>	<pre> SE (<condizione>) ALLORA <istruzione_1> FINE SE </pre>	
Selezione n-aria (chiamata anche scelta n-aria)	<pre> switch (<espressione>) { case <val_1>: { <istruzione_1>; break; } case <val_2>: { <istruzione_2>; break; } case <val_n>: { <istruzione_n>; break; } default: { <istruzione_n+1>; } } </pre>	<pre> NEL CASO CHE (<espressione>) SIA <val_1>: <istruzione_1> <val_2>: <istruzione_2> <val_n>: <istruzione_n> [ALTRIMENTI: <istruzione_n+1>] FINE CASO </pre>	

Nome costruito	In linguaggio C	Pseudocodifica	Flow-chart
Iterazione (con test della condizione in testa o ciclo per "vero")	<pre> while (<condizione>) { <blocco_istruzioni >; } </pre>	<pre> MENTRE (<condizione>) ESEGUI <blocco_istruzioni > FINE MENTRE </pre>	
Iterazione (con test della condizione in coda o ciclo per "falso")	<pre> do { <blocco_istruzioni >; } while NOT(<condizione>); </pre>	<pre> RIPETI <blocco_istruzioni > FINCHE' (<condizione>) </pre>	
Iterazione (con numero di cicli noto a priori)	<pre> for (i=1;i<=n;i++) { <blocco_istruzioni >; } for (i=n;i>=1;i--) { <blocco_istruzioni >; } </pre>	<pre> PER <indice>← <inizio> A <fine> ESEGUI <blocco_istruzioni > INCREMENTA <indice> FINE PER PER <indice>← <fine> INDIETRO A <inizio> ESEGUI <blocco_istruzioni > DECREMENTA <indice> FINE PER </pre>	