Pascal 言語，および C 言語による擬似コード

| | 擬似コード Pascal | 擬似コード C 言語 | 備考 |
|---|---|---|---|
| 逐次構造 | $S_1; S_2; \cdots; S_n$<br>または<br>**begin** $S_1; S_2; \cdots; S_n$ **end**<br>(複合文) | $S_1$;<br>$S_2$;<br>⋮<br>$S_n$; | C 言語で複合文を作るには全体を {} でくくればよい． |
| 選択構造 | **if** P **then** $S_1$ **else** $S_2$ | `if (P) {`<br>$S_1$;<br>`} else {`<br>$S_2$;<br>`}` | 実行文が一つしかなければ {} でくくる必要はない．<br>これより下も同様． |
| | **if** P **then** S | `if (P) {`<br>S;<br>`}` | |
| | case 文 (2c 参照) | `switch (P) {`<br>`case (`$N_1$`):`<br>$S_1$;<br>`break;`<br>`case (`$N_2$`):`<br>$S_2$;<br>`break;`<br>⋮<br>`case (`$N_n$`):`<br>$S_n$;<br>`break;`<br>`default:`<br>$S_d$;<br>`break;`<br>`}` | 左はよく使われる形の一例である．<br>break，default は必要に応じて記述する．<br>詳細は C 言語の教科書を参照すること． |
| 反復構造 | **while** P **do** S | `while (P) {`<br>S;<br>`}` | S の実行の前に P の条件判断を行なう． |
| | **repeat** S **until** P | `do {`<br>S;<br>`} while (P);` | S の実行の後に P の条件判断を行なう． |
| | **for** $v := n_1$ **to** $n_2$<br>**do** S | `for (`$v = n_1; v <= n_2; ++v$`) {`<br>S;<br>`}` | C 言語の for 文はより一般的な反復構造を記述することができる． |
| | **for** $v := n_1$ **downto** $n_2$<br>**do** S | `for (`$v = n_1; v >= n_2; --v$`) {`<br>S;<br>`}` | |