

# Exerciții

1. Implementați o instrucțiune de selecție generalizată numită *case* având sintaxa *case*-ului din limbajul Pascal.
2. Implementați o instrucțiune de selecție generalizată numită *switch* având sintaxa *switch*-ului din limbajul C.
3. \*Implementați o instrucțiune de selecție generalizată numită *if <exp> then <com> elsif <exp> then <com> elsif <exp> then <com> ... else <com>* având sintaxa *if*-ului din limbajul Oberon 2. Notați că după fiecare *if* sau *elsif* există câte-o expresie cu sens de condiție iar după fiecare *<then>* sau *<else>* avem o comandă (instrucțiune). Pot fi oricâte ramuri *elsif*.
4. Explicați dezavantajul unor asemenea implementări bazate pe rescrieri de sintaxă. Indicație: nu uitați să explicați de ce dezavantajul se va manifesta cu ocazia interpretării unor programe lungi.
5. Explicați de ce este mai accesibilă această implementare a FOR-ului. Cum este ea din punct de vedere al modularității întregului sistem ?
6. Găsiți o altă idee pentru a face variabilele secundare din două sau mai multe FOR-uri imbricate să nu se confunde între ele *indiferent* cum ar fi notat programatorul variabilele contor ale buclor.. (Soluția anterior expusă avea dezavantajul că două for-uri imbricate care folosesc incorect variabile contor cu același nume utilizează și pentru calculul limitei superioare variabile cu același nume.)
7. Implementați o instrucțiune *for (<com>,<exp>,<com>) <com>* având sintaxa *for*-ului din limbajul C, folosind tehnica prezentată în acest capitol. Prima comandă va servi la inițializare, expresia *<exp>* va funcționa ca și expresia unei bucle *while*, a doua comandă va asigura trecerea la pasul următor iar ultima comandă este comanda repetată de buclă.