Laboratorul 3. Instrucțiunile limbajului C.

Instrucțiuni condiționale

1. **if ... else**
2. **switch**

if ... else

Poate fi folosită în mai multe forme:

```
if (condition) {
  // instructions
}
```

Evaluează condition şi execută instrucţiunile dintre acolade dacă rezultatul este nenul.

```
if (condition) {
  // instructions
} else {
  // other instructions
}
```

Pentru rezultat nul este executat blocul de instrucţiuni aflat după else.

```
if (condition_1) {
  // instructions_1
} ... else if (condition_n) {
  // instructions_n
}
```

Sunt evaluate pe rând condiţiile şi este executat blocul corespunzător primei condiţii adevărate.

switch

```
switch (expression) {
  case constant_1:
    // instructions_1
    break;
  ...
  default:
    // instructions
}
```

Valoarea expression este **evaluată la un tip întreg** și **comparată cu fiecare constantă**. Este rulat blocul de instrucţiuni al valorii găsite sau blocul aflat după **default** dacă numărul nu este egal cu nici una dintre constante.
După executarea ultimei instrucţiuni dintr-un bloc case, execuţia nu continuă după blocul switch, ci la inceputul următorului bloc case. **Pentru a ieşi din blocul switch, se foloseşte instrucţiunea break.**

Instrucțiuni de repetiție

1. **while**
2. **do ... while**
3. **for**

while

while **execută un bloc de instrucţiuni atât timp cât o anumită condiţie este adevărată**. Forma generală a unui ciclu while este:

```
while (expression) {
  // instructions
}
```

do ... while

do ... while este o instrucţiune repetitivă **similară cu while**, singura diferenţă fiind că **expresia este evaluată după executarea instrucţiunilor**, nu înainte. Astfel, **blocul va fi executat cel puţin o dată**.

```
do {
  // instructions
} while (expression);
```

for

for reprezintă o **formă mai simplă de a scrie un while însoțit de o expresie inițială şi de o expresie de incrementare**. Forma sa este:

```
for (expression_1; expression_2; expression_3) {
  // instructions
}
```

Secvenţa de cod de mai sus este echivalentă cu:

```
expression_1
while (expression_2) {
  // instructions
  expression_3
}
```

În cazul instrucţiunii for, **oricare dintre cele 3 expresii poate lipsi**.

Instrucțiuni speciale

1. **break**
2. **continue**
3. **return**
4. **goto**

break

Pe lângă utilizarea descrisă la instrucţiunea **switch**, instrucțiunea break poate fi folosită pentru a **ieşi forţat dintr-o instrucţiune de repetiţie**. Secvența următoare este echivalentă cu un for:

```
expression_1
for ( ; ; ) {
  if (!expression_2) {
    break; // ieşire forţată din buclă
  }
  expression_3
}
```

continue

continue forţează terminarea iteraţiei curente a buclei si trecerea la iteraţia următoare.

return

return este instrucţiunea de **terminare a funcţiei curente**. Poate fi apelată in forma:

- **return** pentru funcţiile care nu întorc nimic (au void ca tip returnat).
- **return result** pentru funcţiile care întorc o valoare.

goto

goto este o instrucţiune de **salt a execuţiei** ce primeşte ca **parametru o etichetă** și execută instrucţiunile de la acea etichetă.

```
int main() {
  goto label;
  printf("Acest mesaj nu apare la execuţie\n");
label:
  printf("Acest mesaj apare la execuţie\n");
  return 0;
}
```