

Fundamentele programării

Curs 4

Şef lucr.dr.ing. GENGE Béla

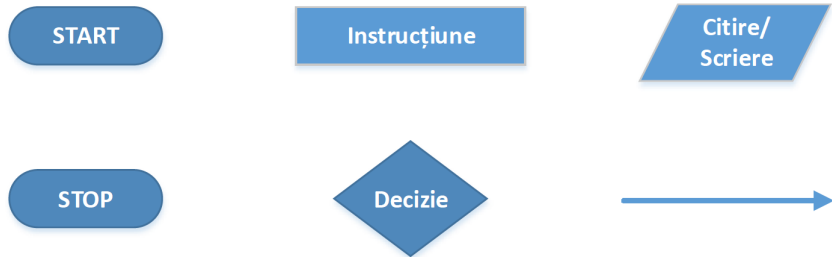
Universitatea “Petru Maior”, Departamentul de Informatică
Tîrgu Mureş, România
bela.genge@ing.upm.ro

- Ciclul de viață a unui program (sau a unui sistem în general) urmează în general următoarele faze:
 - Analiza problemei
 - Proiectarea algoritmului (intrări, ieșiri, secvența pașilor)
 - Implementarea algoritmului
 - Verificarea corectitudinii implementării
 - Testare, mentenanță

- Algoritmii pot fi descriși prin intermediul:
 - Limbajului natural
 - Pseudocodului
 - **Schemelor logice**

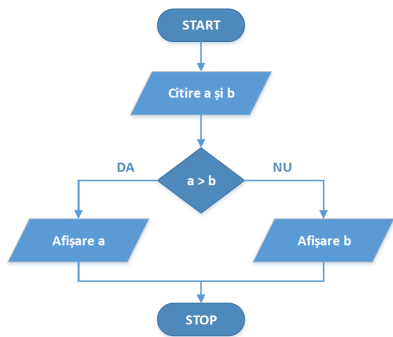
Elementele unei scheme logice

- Blocul de început/sfârșit
- Blocul de instrucțiune, e.g., atribuire, incrementare
- Blocul de decizie, e.g., if
- Blocul de citire/scriere
- Elementul de legătură între blocuri



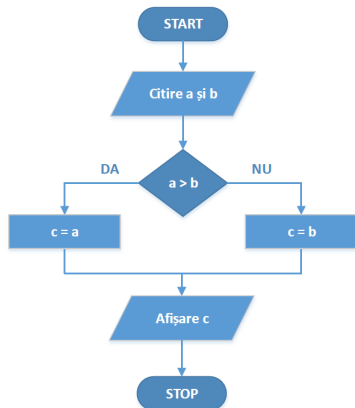
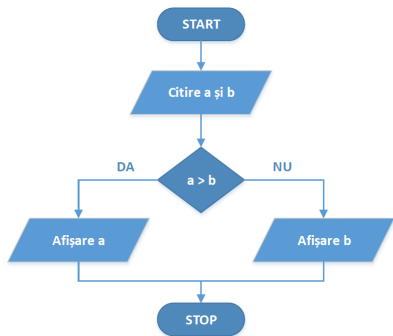
Exemplu - min/max două numere

- Se citesc două numere de la tastatură. Să se determine maximumul dintre ele.

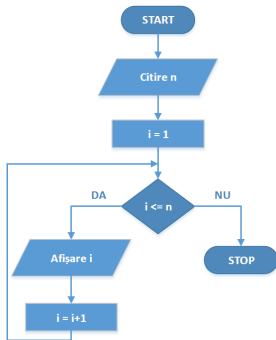


Exemplu - min/max două numere

- Se citesc două numere de la tastatură. Să se determine maximul dintre ele.

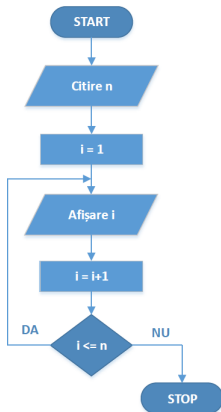
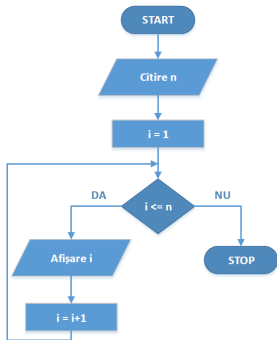


- Cicluri cu test inițial
- Cicluri cu test final
- Exemplu: afișarea numerelor de la 1 la n (citit de la tastatură)



Cicluri

- Cicluri cu test inițial
- Cicluri cu test final
- Exemplu: afișarea numerelor de la 1 la n (citit de la tastatură)



- Să se traseze schema logică și să se implementeze un program 'C' care determină valoarea unei funcții f definită astfel:

$$f(x) = \begin{cases} x, & x > 0 \\ -2x, & x < 0 \\ 1, & x = 0 \end{cases}$$

- Ex. 1: Să se traseze schema logică pentru un program ce determină toate numerele pare de la 1 la n (citit de la tastatură)
- Ex. 2: Să se traseze schema logică pentru un program ce afișează toate literele alfabetului (minuscul)

- Ex. 1: Să se traseze schema logică pentru un program ce determină toate numerele pare de la 1 la n (citit de la tastatură)
- Ex. 2: Să se traseze schema logică pentru un program ce afișează toate literele alfabetului (minuscul)

Instrucțiuni de decizie multiplă

- Reprezintă o generalizare a instrucțiunii de decizie simplă
- Modul de execuție este următorul:
 - Se evaluează expresia
 - Se compară pe rând val. expr cu val_1, ...
 - Dacă val. expr coincide cu una din valori, atunci se execută secv. Dacă secv. e terminată cu break, atunci se trece la instr. următoare după switch. Altfel, se execută secvența următoare.
 - Dacă val. expr. nu coincide cu nici o valoare se execută secv. default.

```
switch(expr)
{
case val_1: secventa_1; [break;]
case val_2: secventa_2; [break;]
...
case val_n: secventa_n; [break;]

[default: secventa]
}
```

Exemplu

- Să se determine dacă un caracter citit de la tastatură este o vocală (alfabetul Englezesc)

Exemplu de cod

```
scanf("%c", &c);
switch(toupper(c))
{
case 'A':
case 'E':
case 'I':
case 'O':
case 'U': printf("Caracterul este o vocală\n"); break;

default: printf("Caracterul este o consoană\n");
}
```

Exemple - care este corect (pentru aceeași problemă)?

Exemplu 1

```
switch(toupper(c)){
case 'A': break;
case 'E': break;
case 'I': break;
case 'O': break;
case 'U': printf("Caracterul este o vocală\n"); break;}
```

Exemplu 2

```
switch(toupper(c)){
case 'A': printf("Caracterul este o vocală\n"); break;
case 'E': printf("Caracterul este o vocală\n"); break;
case 'I': printf("Caracterul este o vocală\n"); break;
case 'O': printf("Caracterul este o vocală\n"); break;
case 'U': printf("Caracterul este o vocală\n"); break;}
```

Exemple - este corect?

Exemplu

```
switch(toupper(c)){  
case 'A'==c&&'E'==c:  
case 'I'==c&&'O'==c:  
case 'U': printf("Caracterul este o vocală\n"); break;}
```

- Să se implementeze secvența de cod ce verifică dacă un caracter citit de la tastatură este:
 - Cifră
 - Vocală
 - Semn de punctuație
- Într-o primă fază se va construi schema logică.
- Implementarea se va realiza atât cu `if` cât și cu `switch`.