

# Aplicatia Pseudocod. Structuri de control

## BREVIAR TEORETIC

Structurile repetitive permit comanda execuției repetate a unei secvențe funcție de respectarea unei anumite condiții de oprire. Există trei forme distincte de repetiții:

- cu test inițial,
- cu test final și
- cu contor.

Structura repetitivă repeta ... cat timp / pana cand

```
[ repeta
  <secvența>
  cât timp <condiție> ] [ repeta
  <secvența>
  până când <condiție> ]
```

Structura repetitiva cu contor pentru

```
pentru contor = val_inicial, val_finala, pasexecuta
  Set_instructiuni
@
```

*contor* este o variabilă care parcurge intervalul (*val\_inițială*, *val\_finala*) la fiecare iterație modificându-se cu *pas*.

## PROBLEME REZOLVATE

Calculați următoarea sumă:  $1 + 4 + 7 + 10 + 13 + \dots + N$

```
start
int n, s, contor
repetă
  scrie "\n Dati valoarea lui n "
  citește n
cat timp n<=0
s = 0
pentru contor = 1, n, 3 executa
```

```

s = s + contor
@
scrie "\n Suma numerelor este : ", s
stop

```

### PROBLEME PROPUSE SPRE REZOLVARE

1. Să se realizeze o aplicație care citește  $n$  numere de la tastatură ( $n$  să fie mai mic decât 100) și afișează suma lor. La realizarea aplicației se va ține cont de faptul că un utilizator neinstruit poate introduce și valori negative sau valori mai mari decât 100 pentru  $n$ .
2. Să se realizeze o aplicație care citește  $n$  numere de la tastatură ( $n$  să fie mai mic decât 100) și afișează numărul de elemente negative și produsul celor pozitive. La realizarea aplicației se va ține cont de faptul că un utilizator neinstruit poate introduce și valori negative sau valori mai mari decât 100 pentru  $n$ .

### BREVIAR TEORETIC: VECTORI (ȘIRURI DE NUMERE)

Vectorii sunt reprezentați de mai multe variabile localizate la adrese succesive de memorie și care poartă o denumire formată din:

- o parte comuna (denumirea vectorului);
- o denumire specifică, dată de poziția față de începutul șirului, încadrată de parantezele [].

Declarația unui vector:

```
int vector[20]
```

unde:

- int - tipul elementelor din vector
- vector - denumirea comuna a elementelor
- 20 - numărul maxim de elemente din vector

Elementele din vector vor fi folosite cu denumirile: vector[0], vector [1] etc.:

vector[0]	vector[1]	vector[2]	vector[3]	.....	vector[19]

Se observă că primul element din vector este referit ca vector[0], adică indicii elementelor din vector încep de la 0 și se termina cu 19 (adică nr\_maxim\_de\_elemente -1).

Pentru a da valori primelor 4 elemente din vector ar putea fi scris codul:

```

start
int vect[20]
scrie " Dati elementul 1 "

```

```
citeste vect[0]
scrie " Dati elementul 2 "
citeste vect[1]
scrie " Dati elementul 3 "
citeste vect[2]
scrie " Dati elementul 4 "
citeste vect[3]

scrie " Elementele introduse sunt: "
scrie vect[0], " ", vect[1], " ", vect[2], " ", vect[3], " "

stop
```

Această variantă nu este recomandată din mai multe motive:

- dacă am un număr mare de elemente (de exemplu 100) trebuie să scriu prea multe linii de cod;
- dacă utilizatorul vrea să introducă un număr variabil de elemente (la o execuție a programului vrea 5 elemente, iar la alta execuție 50 de elemente) codul nu mai poate fi scris după exemplul de mai sus.

Varianta mai elegantă și care va fi folosită:

```
start

int n, i
int vect[20]

scrie "Dati numarul de elemente "
citeste n
pentru i= 0, n-1 executa
    scrie " Dati elementul cu indicele ", i, " "
    citeste vect[i]
@

scrie " Elementele introduse sunt: "
pentru i= 0, n-1 executa
    scrie vect[i], " "
@

stop
```

### PROBLEME PROPUSE SPRE REZOLVARE

1. Afișați elementele unui vector în ordine inversa.
2. Afișați elementele unui vector care se afla pe pozițiile pare.
3. Afișați elementele impare ale unui vector.

### BREVIAR TEORETIC: MATRICI (ȘIRURI DE NUMERE)

**DECLARARE****Sintaxa generală:**

```
int matr[20][10]
```

**unde:**

int - tipul elementelor din matrice  
matrice - denumirea comuna a elementelor  
20 - numărul maxim de linii din matrice  
10 - numărul maxim de coloane din matrice

O linie din matrice este de fapt un vector cu denumirea `matrice[indice]`. Prin urmare putem considera că matricea este de fapt un șir de vectori: `matr[0]`, `matr[1]`, ... `matr[19]`.

Dacă numele unui vector este `matr[0]`, după cum am explicat mai sus, elementele primei linii din matrice sunt de fapt elementele vectorului `matr[0]`, și deci sunt accesate cu denumirile:

`matr[0][0]`, `matr[0][1]`, `matr[0][2]`, ..., `matr[0][9]`

Elementele liniei 2 din matrice sunt de fapt elementele vectorului `matr[1]` și au denumirile:

`matr[1][0]`, `matr[1][1]`, `matr[1][2]`, ..., `matr[1][9]`.

În modul acesta se obțin denumirile tuturor elementelor din matrice.

Pentru a citi o matrice de 2 linii și 3 coloane am putea scrie codul:

```
start
int matr[2][3]

scrie " Dati elementele primei linii"
scrie " Dati elementul 1 de pe prima linie"
citeste matr[0][0]
scrie " Dati elementul 2 de pe prima linie"
citeste matr[0][1]
scrie " Dati elementul 3 de pe prima linie"
citeste matr[0][2]

scrie " Dati elementele de pe linia a doua"
scrie " Dati elementul 1 de pe linia a doua "
citeste matr[1][0]
scrie " Dati elementul 2 de pe linia a doua "
citeste matr[1][1]
scrie " Dati elementul 3 de pe linia a doua "
citeste matr[1][2]

scrie " Elementele matricii sunt: "
scrie "\n ", matr[0][0], " ", matr[0][1], " ", matr[0][2]
scrie "\n ", matr[1][0], " ", matr[1][1], " ", matr[1][2]

stop
```

Sau, ținând cont de observațiile de la vectori, am putea scrie codul:

```
start

int j, nc
int matr[20][10]

scrie "Dati numarul de coloane ale matricii "
citeste nc

scrie " Dati elementele primei linii "
pentru j = 0, nc-1 executa
    citeste matr[0][j]
@

scrie " Dati elementele de pe a doua linie "
pentru j = 0, nc-1 executa
    citeste matr[1][j]
@

scrie " Elementele introduse sunt: "
pentru j = 0, nc-1 executa
    scrie matr[0][j], " "
@

pentru j = 0, nc-1 executa
    scrie matr[1][j], " "
@

stop
```

Sau - dacă vrem să fie într-adevăr flexibil - atât din punctual de vedere al numărului de linii cât și din punctual de vedere al numărului de coloane, vom scrie codul:

```
start

int nl, nc, i, j
int matr[20][10]

scrie "Dati numarul de linii ale matricii "
citeste nl
scrie "Dati numarul de coloane ale matricii "
citeste nc

pentru i = 0, nl-1 executa
    scrie " Dati elementele liniei cu indicele ", i, " "
    pentru j = 0, nc-1 executa
        citeste matr[i][j]
    @
@

scrie " Elementele introduse sunt: "
pentru i = 0, nl-1 executa
    scrie "\n"
    pentru j = 0, nc-1 executa
        scrie matr[i][j], " "
    @
```

---

@

stop

### PROBLEME PROPUSE SPRE REZOLVARE

1. Afișarea diagonalei principale din matrice.
2. Suma elementelor din matrice.
3. Afișarea sumelor elementelor de pe linii.