

Tema 3

(curs 3 - 52)

Ex1: Pentru fiecare din următoarele relații explicați de ce se respectă/nu se respectă proprietățile de reflexivitate, simetrie, asimetrie și tranzitivitate

a) "EsteFrateleLui" pentru un set de persoane

b) $R = \{ \langle x, y \rangle \mid x^2 = y^2 \}$ pentru numere reale

c) $R = \{ \langle x, y \rangle \mid x \bmod y = 0 \}$ pentru $x \in \{1, 2, 3, 4\}$

a) Reflexivitate : nu poate fi reflexiva, deoarece o persoana nu poate fi propriul frate

Simetrie : da, daca x este fratele lui y, atunci si y este fratele lui x

Asimetrie : nu, deoarece este simetrica relatia

Tranzitivitate : da, deoarece daca x este fratele lui y

\Rightarrow atunci x este fratele lui z de asemenea

si y este fratele lui z

Reflexivitate : da, deoarece pentru orice x numar real, $\langle x, x \rangle \Rightarrow x^2 = x^2$

Simetrie : da, deoarece daca $x^2 = y^2 \Rightarrow y^2 = x^2$, deci $\langle y, x \rangle$ apartine R

Antisimetrie : nu

Tranzitivitate : da, deoarece daca $x^2 = y^2$

$\Rightarrow x^2 = z^2$ si $\langle x, z \rangle$ apartine R

si $y^2 = z^2$

Reflexivitate : da, orice nr este divizibil cu el insusi

Simetrie : nu, daca $x \bmod y == 0$, atunci nu inseamna neaparat ca si $y \bmod x == 0$

ex : $4 \bmod 2 == 0$, dar $2 \bmod 4 \neq 0$

Antisimetrie : da, daca $x \bmod y == 0$ si

$\Rightarrow x == y$

daca $y \bmod x == 0$

Tranzitivitate : da, daca $x \bmod y == 0$

$\Rightarrow x \bmod z == 0$

si $y \bmod z == 0$

Ex3: Să se definească un TDA pentru o mulțime cu elemente întregi

Ex4: Rescrieți funcția factorial astfel încât să nu fie recursivă

~~Ex5: Demonstrați prin inducție formulele din secțiunea "Sume și recurențe"~~

~~Ex6: Să se demonstreze folosind tehnica inducției puternice că orice număr natural n (strict pozitiv) poate fi scris sub forma~~

~~$n = 2^i \cdot j$, unde i este un număr întreg ≥ 0 și j este un număr natural impar~~

ex 3 :

```
#include <stdio.h>
#include <stdlib.h>
```

```
typedef struct{
    int *elemente;
    int dimensiune_curenta;
    int capacitate;
}multimeElementeIntregi;
```

```
multimeElementeIntregi creareMultime(int capacitateMaxima);
```

```
void elibereazaSpatiuMultime(multimeElementeIntregi *multime);
```

```
void adaugaInMultime(multimeElementeIntregi *multime, int elementDeAdaugat);
```

```
void stergeDinMultime(multimeElementeIntregi *multime, int elementDeSters);
```

```
void afisare(multimeElementeIntregi *multime);
```

ex 4 :

recursiv

```
long factorial(int n)
{
    if (n == 1)
        return 1; //conditia de oprire
    else
        return(n * factorial(n - 1));
}
```

iterativ

```
long factorialIterativ(int n){
    long rezultat = 1;

    for(int i = 2; i <= n; i++){
        rezultat *= i;
    }

    return rezultat;
}
```