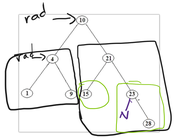


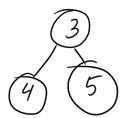
inordine (S R D)

1,4,9 / SS ; 10 ; 15, 21, 23,28 / SD

Funcție recursivă:

```
void inordine (rad)
  if rad ≠ NULL then
      call inordine (rad → stang)
      write rad → info
      call inordine (rad → dept)
  endif
```

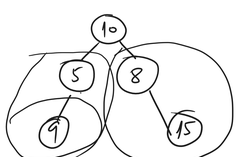
preordine 3,4,5.



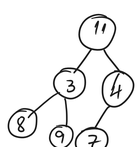
inordine 4,3,5

postordine SDR

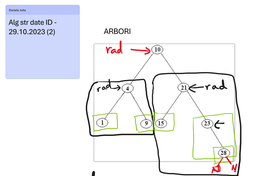
4 / SS ; 5 / SD ; 3 / R.



9, 5, 15, 8, 10.



8,9,3, 7,4, 11.



Fiecare nod


```
Struct NodArb
{ T info;
  NodArb *stang, *dept;
};
```

Parcurgerea arborilor:

preordine - RSD

10, 4, 1, 9 / SS ; 21, 15 / SD ; 23, 28

Funcție recursivă

```
void preordine (rad)
  //rad = pointer la rădăcina
  // arborelui care trebuie parcurs
  if rad ≠ NULL then
      write rad → info
      // parcurgere subarb. stang
      call preordine (rad → stang)
      call preordine (rad → dept)
  endif
```