

Nume, Prenume, grupa _____

Se dă tipul de date:

data Tree23 a = Empty

| Node2 (Tree23 a) a (Tree23 a)

| Node3 (Tree23 a) a (Tree23 a) a (Tree23 a)

| Node4 (Tree23 a) a (Tree23 a) a (Tree23 a) a (Tree23 a) ---acest nod

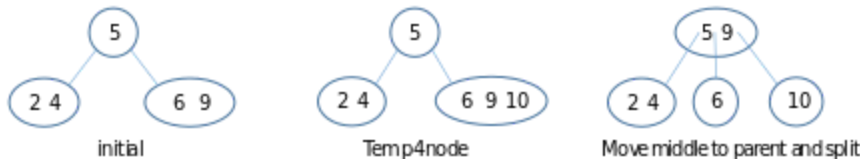
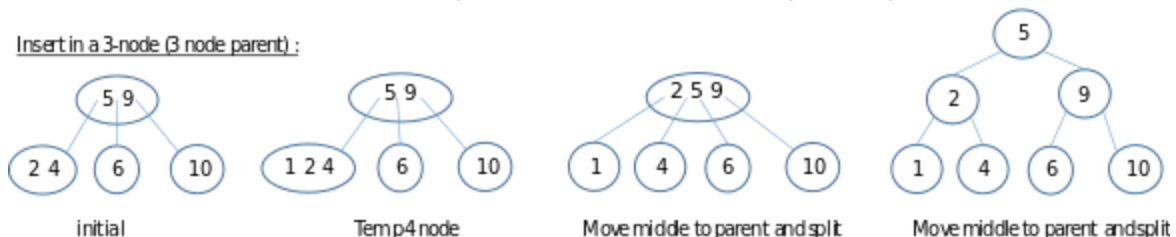
este folosit ca nod auxiliar pentru funcția insert (nu afectează restul exercițiilor)

Proprietățile unui 2-3 Arbore:

- Fiecare nod are una sau două valori;
- Un nod cu o valoare este fie frunza (nodul 100 din exemplu), fie are exact doi fii (nodul 39); Valorile din subarborele stâng < valoarea din nod < valorile din subarborele drept;
- Un nod cu două valori este fie frunză (nodul (10 20)) fie are exact trei fii. Valorile din subarborele stâng < valorile din primul nod < valorile din subarborele din mijloc < valorile din subarborele drept;
- Toate frunzele sunt pe același nivel.

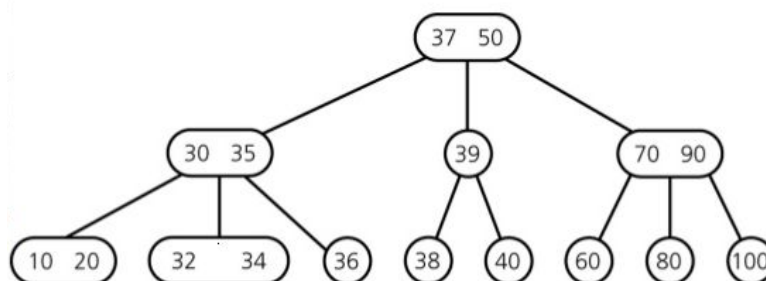
Inserarea lui x într-un 2-3 Arbore se face astfel:

- Dacă arborele e gol se crează un nod și se pune valoarea x în nod;
- Altfel, se găsește un nod frunză unde trebuie inserat x și se adaugă x în el
- Dacă nodul avea deja două valori, se va forma un nod de tip 4;
- Un nod de tip 4 va fi modificat astfel: valoarea mediană din nod va fi adăugată în nodul părinte, valoarea minimă va forma un nod de tip 2 și va reprezenta subarborele din mijloc al părintelui, iar valoarea maximă va forma un nod de tip 2 și va reprezenta subarborele drept al părintelui. În cazul în care nodul de tip 4 este rădăcina arborelui, valoarea mediană va forma un nod de tip 2 și va deveni noua rădăcină, iar valoarea minimă și cea maximă vor forma subarborii stâng și drept.

Insert in a 2-node :Insert in a 3-node (2 node parent) :Insert in a 3-node (3 node parent) :

Nume, Prenume, grupa _____

Exemplu:



myTree :: Tree23 Int

```
myTree = (Node3 (Node3 (Node3 Empty 10 Empty 20 Empty) 30 (Node3 Empty 32 Empty 34
Empty) 35 (Node2 Empty 36 Empty)) 37 (Node2 (Node2 Empty 38 Empty) 39 (Node2 Empty 40
Empty)) 50 (Node3 (Node2 Empty 60 Empty) 70 (Node2 Empty 80 Empty) 90(Node2 Empty 100
Empty)))
```

Cerințe:

1. (1 punct)
 - a. Să se scrie un modul Tree23 care definește tipul de date parametrizat Tree23 al 2-3 arborilor sortați, și exportă tipul de date **Tree23** și funcțiile **contains**, **toList**, **count3**, **insert**, **fromList** și **myTree**.
 - b. Să se scrie un modul Main, executabil, care folosește funcțiile din modulul Tree23 pentru a citi de la tastatură o listă de numere separate prin spațiu, aflate pe aceeași linie, inserează numerele în arborele myTree și afișează numărul de noduri de tip 3 care exista în arbore.
2. Să se implementeze următoarele funcții:
 - a. (2 puncte) **contains**: primește ca parametru un element de tip **a** și un element de tip Tree23 a și verifică dacă elementul se găsește în arbore.
contains 39 myTree = True
contains 7 myTree = False
 - b. (2 puncte) **toList**: pentru un 2-3 arbore întoarce lista formată elementele arborelui în ordine crescătoare, parcurgând lista în inordine (stânga-rădăcină-dreapta pentru noduri de tip 2 și stânga-rădăcină1-mijloc-rădăcină2-dreapta pentru noduri de tip 3)
toList myTree = [10,20,30,32,34,35,36,37,38,39,40,50,60,70,80,90,100]
 - c. (2 puncte) **count3**: numără câte noduri de tip 3 are arborele.
count3 myTree = 5
 - d. (1 puncte) **insert**: adaugă un element dat într-un arbore astfel încât arborele să rămână echilibrat.
 - e. (1 puncte) **fromList**: dată fiind o listă de elemente creează un 2-3 arbore de căutare care le conține.

1 punct din oficiu. Timp de lucru 50 minute**Atenție: anumite funcții necesită constrângeri de tipuri**