1. Interclasarea a doua liste ordonate produce o singura lista ordonata cu toate valorile din listele initiale. Scrieti urmatoarele functii pentru liste inlantuite ordonate fara santinela:

* add : adaugarea unei valori la sfarsitul unei liste (recursiv);
* ins : inserarea unei valori date intr-o lista ordonata (recursiv);
* intercl1 : interclasarea recursiva a 2 liste ordonate folosind ’’ins” (cu 2 argumente ; rezultatul interclasarii inlocuieste prima lista);
* intercl2: interclasarea a 2 liste ordonate folosind functia ’’add” si crearea unei liste intermediare (cu 2 argumente).

Ex: a={2,4,6}; b={1,2,3,5}; a(c) = {1,2,2,3,4,5,6}

1. O lista inlantuita contine in fiecare nod 2 intregi x si y si este ordonata in principal dupa x si in secundar dupa y (nodurile au aceeasi valoare si sunt ordonate dupa y).

Ex: (2,1), (2,3), (2,5), (3,2), (4,2), (4,6), (5,4).

Scrieti functii pentru operatii cu astfel de liste:

* Adaugarea unei perechi (x,y) la o lista ordonata cu mentinere ordine;
* Crearea unei liste b de perechi (x,y) cu nodurile care au aceeasi valoare x dintr-o lista data;
* Verificarea daca o lista de perechi (x,y) este ordonata dupa x si apoi dupa y.

1. O colectie de n multimi este un vector de pointeri la liste inlantuite neordonate. Scrieti:

* Definiti tipul ’’set” (o multime) si ’’sets” (colectie de multimi);
* Functie care face reuniunea a doua multimi:

add2(set &a,set b);

* Functie care primeste o colectie de multimi liste si creaza o multime cu reuniunea tuturor multimilor din colectie:

addAll(sets c, set &r); // r- reuniune

Ex: {3,1,4,2}, {2,3,5,1,7}, {1,5,4,6,3}

r = {1,2,3,4,5,6,7}