1) Sa se defineasca doua predicate: lungimepara(+Lista) si lungimeimpara(+Lista), care verifica daca numarul de elemente al unei liste este par sau impar.
Indicatie: In definirea predicatului lungimepara trebuie sa ne referim la predicatul lungimeimpara, iar in definirea predicatului lungimeimpara trebuie sa ne referim la predicatul lungimepara.

2) Sa se defineasca predicatul max(+X,+Y,-Max), unde Max reprezinta maximul dintre X si Y. Folosind predicatul max, sa se defineasca predicatul maxlista(+Lista,-MaxLista), astfel incat MaxLista sa intoarca cel mai mare element al listei Lista.

3) Sa se defineasca predicatul separa(+Lista,-ListaPozitive,-ListaNegative) care imparte lista Lista in doua subliste, una cu numerele pozitive, iar cealalta cu numerele negative.

4) Sa se defineasca predicatul imparte(+Lista,-Lista1,-Lista2), care imparte lista Lista in doua subliste, prima va contine elementele de pe pozitiile pare, iar a doua va contine elementele de pe pozitii impare.

5) Sa se scrie predicatul interclasare(+Lista1,+Lista2,-ListaRezultat), care realizeaza interclasarea a doua liste sortate crescator. Daca avem elemente comune in cele doua liste, acestea vor apare o singura data in lista rezultat ordonata crescator.

6) Sa scrie predicatul sterge(+X,+Lista,-ListaRezultat), care sterge toate aparitiile elementului X din lista Lista si pune rezultatul in ListaRezultat.

7) Sa se scrie predicatul eliminduplicate(+Lista,-ListaRezultat), care elimina toate duplicatele unui element dintr-o lista.
Indicatie: In definirea predicatului eliminduplicate trebuie folosit predicatul predefinit member(+X,+Lista) care testeaza apartenenta unui element la o lista (este, de fapt, predicatul apartine pe care l-am definit saptamana trecuta).