**LABORATOR 1** (cerinta nr. 15)

Să se implementeze în C++ un anumit container de date (TAD) folosind o anumită reprezentare (indicată) *și un vector dinamic* ca structură de date. Se va folosi implementare proprie pentru vectorul dinamic.

**TAD Dictionar Ordonat** – reprezentare sub forma unui vector dinamic de perechi (cheie, valoare) ordonate în raport cu cheile și cu o relație de ordine ℜ între chei .

**LABORATOR 2** (cerinta nr. 6)

Să se implementeze în C++ un anumit container de date (TAD) folosind o anumită reprezentare (indicată) și *o listă înlănțuită* (LSI – lista simplu înlănțuită, LDI – lista dublu înlănțuită) ca structură de date, cu alocare dinamică pentru memorarea înlănțuirilor.

**TAD Colecție cu elemente de tip comparabil** (TElement=TComparabil)– reprezentare folosind o LDI cu elementele ordonate în raport cu relația de ordine ℜ=≤.

**LABORATOR 3** (cerinta nr. 14)

Să se implementeze în C++ un anumit container de date(TAD) folosind o anumită reprezentare (indicată) și o listă înlănțuită(LSI –lista simplu înlănțuită, LDI –lista dublu înlănțuită) ca structură de date, cu reprezentarea înlănțuirilor pe tablou dinamic.

**TAD Lista** (interfața cu TPoziție=Iterator) –reprezentare folosind o LDI.

**LABORATOR 4** (cerinta nr. 25)

Să se implementeze în C++ un anumit container de date(TAD) folosind o anumită reprezentare (indicată) și o tabelă de dispersie(TD) ca structură de date, cu o anumită metodă (indicată) pentru rezolvarea coliziunilor:

−Liste independente

−Liste întrepătrunse

−Adresare deschisă

TAD MultiDicționar Ordonat –memorarea tuturor perechilor de forma (cheie, valoare), reprezentare sub forma unei TD cu rezolvare coliziuni prin liste întrepătrunse.