

## Lista – implementare secvențială (statică, cu vectori)

```
#include <iostream>
using namespace std;

//dimensiunea maxima a vectorului
const int MAX_SIZE=100;

//declaram lista
int LISTA [MAX_SIZE];

//crearea unei liste vide
int lungime=0;

// inserare element la sfarsitul listei
void inserare_final(int LISTA[], int &lungime, int valoare)
{
    if (lungime>=MAX_SIZE)
    {
        cout<<"Lista este plina!"<<endl;
        return;
    }

    LISTA[lungime]=valoare;
    lungime++;
}

// inserare element in lista, pe o anumita pozitie
void inserare(int LISTA[], int &lungime, int pozitie, int valoare)
{
    if (lungime>=MAX_SIZE || pozitie<0 || pozitie>lungime)
    {
        cout<<"Pozitie invalida sau lista este plina!"<<endl;
        return;
    }

    // deplasam cu o pozitie la dreapta toate elementele din lista
    for (int i=lungime; i>pozitie; i--)
        LISTA[i] = LISTA[i - 1];

    //inseram elementul pe pozitia ceruta
    LISTA[pozitie]=valoare;

    //creste lungimea listei
    lungime++;
}
```

```

// eliminare element de pe o anumita pozitie
void eliminare (int LISTA[], int &lungime, int pozitie)
{
    if (pozitie<0 || pozitie>=lungime)
    {
        cout<<"Pozitie inexistent!"<<endl;
        return;
    }

    // deplasam cu o pozitie la stanga toate elementele din lista
    // incepand cu pozitia elementului care se elimina
    for (int i=pozitie; i<lungime-1; i++)
        LISTA[i]=LISTA[i+1];

    //scade lungimea listei
    lungime--;
}

// afisare lista
void afisare(int LISTA[], int &lungime)
{
    cout<<"Lista ["<<lungime<<"]: ";
    for (int i=0; i<lungime; i++)
    {
        cout<<LISTA[i]<<" ";
    }
    cout<<endl;
}

int main() {
    //am declarat deja global lista si lungimea ei

    inserare_final(LISTA, lungime, 10);
    inserare_final(LISTA, lungime, 20);
    inserare_final(LISTA, lungime, 30);
    inserare_final(LISTA, lungime, 40);
    afisare(LISTA, lungime); // Lista[4]: 10 20 30 40

    inserare(LISTA, lungime, 2, 25); // inseram valoarea 25 pe pozitia 2
    afisare(LISTA, lungime); // Lista[5]: 10 20 25 30 40

    eliminare(LISTA, lungime, 3); // eliminam elementul de pe pozitia 3 (care era 30)
    afisare(LISTA, lungime); // Lista[4]: 10 20 25 40
    return 0;
}

```