

**LABORATOR NR. 2****CHANCHIAN ARMIN ANDREI - INFO - AN2 -IF****PROBLEME:**

1. Sa se scrie un program C++ care implementeaza o lista de intregi, alocata secvential. Operatiile de baza se vor face prin intermediul functiilor. Vectorul va fi alocat cu dimensiunea  $N = \text{nr max posibil de elemente}$ .

La rulare trebuie sa se poata alege dintr-un meniu una din operatiile:

1. **Introducere elem** – se va cere valoarea de introdus si pozitia noului elem
2. **Stergere elem** - se va cere pozitia elem de sters; va afisa elementul sters
3. **Accesare elem** - se va cere pozitia elem
4. **Modificare elem** - se va cere pozitia elem si noua valoare
5. **Creare lista** - va permite crearea unei liste cu mai multe elem, inserate in ordine: primul, al doilea, ...elem fiind introduse de la tastatura
6. **Afisare lista** – afisarea elem listei

Orice alt input---exit iesirea din program

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

const int N = 100;
int n;
int lista[N];

void introducere() {
    cout << "Introduceti valoarea elementului pe care doriti sa il introduceti" << endl;
    int poz, k;

    cin >> k;

    cout << endl << " si pozitia acestuia";

    cin >> poz;

    if (n == N) {
        cout << "OVERFLOW";
    }
    else if (poz > n || poz < 0) {
        cout << "Pozitia este invalida";
    }
    else {
        for (int i = n - 1; i >= poz - 1; i--) {
            lista[i + 1] = lista[i];
        }
        lista[poz - 1] = k;
        n++;
    }
}

void createList() {
    cout << "Introduceti numarul de elemente: " << endl;

    cin >> n;

    cout << "Introduceti elementele: " << endl;
    for (int i = 0; i < n; i++) {
        cin >> lista[i];
    }
}

void printList() {
    for (int i = 0; i < n; i++) {
        cout << lista[i] << " ";
    }
    cout << endl;
}
```

```

}

void stergere() {
    int poz;
    cout << endl << "Introduceti pozitia elementului pe care doriti sa il stergeti: ";

    cin >> poz;

    if (n == 0) {
        cout << "UNDERFLOW";
    }
    else if (n >= 0 || poz < 0) {
        cout << "Pozitia este invalida";
    }
    else {
        cout << "Valoarea elementului sters este: " << lista[poz] << endl;

        for (int i = poz; i < n - 1; i++) {
            lista[i] = lista[i + 1];
        }
        n--;
    }
}

void accesare() {
    cout << "Introduceti pozitia elementului: ";

    int poz;

    cin >> poz;

    if (poz < 0 || poz > n) {
        cout << "Pozitie in afara listei";
    }
    else {
        cout << lista[poz];
    }
}

void modificare() {
    cout << "Introduceti pozitia elementului care urmeaza sa fie modificat si noua valoare: ";
    int poz, newValue;

    cin >> poz >> newValue;

    if (poz < 0 || poz > n) {
        cout << "Pozitie in afara listei";
    }
    else {
        lista[poz] = newValue;
    }
}

```

```

        cout << lista[poz];
    }
}

int main() {
    int input;

    cout << "Introduceti o comanda: " << endl;
    cin >> input;

    for (;;) {
        switch (input) {
            case -1:
                exit(0);
            case 1:
                // Inserare element
                introducere();
                break;
            case 2:
                // Stergere element
                stergere();
                break;
            case 3:
                // Accesare element
                accesare();
                break;
            case 4:
                // Modificare element
                modificare();
                break;
            case 5:
                // Afisare lista
                printList();
                break;
            case 6:
                // Citire lista
                createList();
                break;
            default:
                cout << "Comanda aleasa nu exista" << endl;
                break;
        }
    }

    cout << "Introduceti alta comanda: " << endl;
    cin >> input;

    return 0;
}

```