**PRACTICAL 16**

**HARSH VEKARIYA**

**23162121024**

**BATCH 24**

**CODE:**

#include <iostream>

#include <vector>

#include <deque>

#include <list>

#include <set>

#include <map>

#include <stack>

#include <queue>

#include <algorithm>

using namespace std;

void vectorModule() {

    vector<int> vec;

    vec.push\_back(10);

    vec.push\_back(20);

    vec.push\_back(30);

    cout << "Vector size: " << vec.size() << endl;

    cout << "Vector elements: ";

    for (auto it = vec.begin(); it != vec.end(); ++it) {

        cout << \*it << " ";

    }

    cout << endl;

    vec.pop\_back();

    cout << "After pop\_back(), Vector size: " << vec.size() << endl;

}

void dequeModule() {

    deque<int> dq;

    dq.push\_back(10);

    dq.push\_front(5);

    dq.push\_back(15);

    cout << "Deque size: " << dq.size() << endl;

    cout << "Deque elements: ";

    for (auto it = dq.begin(); it != dq.end(); ++it) {

        cout << \*it << " ";

    }

    cout << endl;

    dq.pop\_back();

    cout << "After pop\_back(), Deque size: " << dq.size() << endl;

}

void listModule() {

    list<int> li;

    li.push\_back(10);

    li.push\_front(5);

    li.push\_back(15);

    cout << "List size: " << li.size() << endl;

    cout << "List elements: ";

    for (auto it = li.begin(); it != li.end(); ++it) {

        cout << \*it << " ";

    }

    cout << endl;

    li.pop\_back();

    cout << "After pop\_back(), List size: " << li.size() << endl;

}

void setModule() {

    set<int> s;

    s.insert(10);

    s.insert(20);

    s.insert(10); // Duplicate, won't be added

    cout << "Set size: " << s.size() << endl;

    cout << "Set elements: ";

    for (auto it = s.begin(); it != s.end(); ++it) {

        cout << \*it << " ";

    }

    cout << endl;

    s.erase(20);

    cout << "After erase(), Set size: " << s.size() << endl;

}

void mapModule() {

    map<string, int> mp;

    mp["apple"] = 10;

    mp["banana"] = 20;

    cout << "Map size: " << mp.size() << endl;

    cout << "Map elements: ";

    for (auto it = mp.begin(); it != mp.end(); ++it) {

        cout << it->first << ":" << it->second << " ";

    }

    cout << endl;

    mp.erase("banana");

    cout << "After erase(), Map size: " << mp.size() << endl;

}

void stackModule() {

    stack<int> s;

    s.push(10);

    s.push(20);

    s.push(30);

    cout << "Stack size: " << s.size() << endl;

    cout << "Stack top: " << s.top() << endl;

}

void queueModule() {

    queue<int> q;

    q.push(10);

    q.push(20);

    q.push(30);

    cout << "Queue size: " << q.size() << endl;

    cout << "Queue front: " << q.front() << endl;

    cout << "Queue back: " << q.back() << endl;

}

int main() {

    int choice;

    do {

        cout << "\n1. Vector Module\n";

        cout << "2. Deque Module\n";

        cout << "3. List Module\n";

        cout << "4. Set Module\n";

        cout << "5. Map Module\n";

        cout << "6. Stack Module\n";

        cout << "7. Queue Module\n";

        cout << "8. Exit\n";

        cout << "Enter your choice: ";

        cin >> choice;

        switch (choice) {

            case 1:

                vectorModule();

                break;

            case 2:

                dequeModule();

                break;

            case 3:

                listModule();

                break;

            case 4:

                setModule();

                break;

            case 5:

                mapModule();

                break;

            case 6:

                stackModule();

                break;

            case 7:

                queueModule();

                break;

            case 8:

                cout << "Exiting...";

                break;

            default:

                cout << "Invalid choice!" << endl;

        }

    } while (choice != 8);

    return 0;

}

**OUTPUT :**



