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

class Linear {
public:
    int arr[100];
    int size;

    void accept();
    int Search(int target);
    int countRep(int target);
    void Occurrence(int target);
} L;

void Linear::accept() {
    cout << "Enter the size of the array (max 100): ";
    cin >> size;
    cout << "Enter the elements of the array: ";
    for (int i = 0; i < size; i++) {
        cin >> arr[i];
    }
    cout << "The elements of the array are: ";
    for (int i = 0; i < size; i++) {
        cout << arr[i] << "\n";
    }
}

int Linear::Search(int target) {
    for (int i = 0; i < size; i++) {
        if (arr[i] == target) {
            return i;
        }
    }
    return -1;
}



int Linear::countRep(int target) {
    int count = 0;
    for (int i = 0; i < size; i++) {
        if (arr[i] == target) {
            count++;
```

```
        }
    }
    return count;
}

void Linear:: Occurrence(int target) {
    int first = -1, last = -1;
    for (int i = 0; i < size; i++) {
        if (arr[i] == target) {
            if (first == -1) {
                first = i;
            }
            last = i;
        }
    }
    if (first != -1) {
        cout << "First occurrence of element " << target << " is at index " << first << "\n";
        cout << "Last occurrence of element " << target << " is at index " << last << "\n";
    } else {
        cout << "Element not found in the array.\n";
    }
}

int main() {
    int choice, target;

    while (true) {
        cout << "\nMenu:\n";
        cout << "1. Accept array elements\n";
        cout << "2. Search for an element\n";
        cout << "3. Count repetitions of an element\n";
        cout << "4. Display first and last occurrence of an element\n";
        cout << "5. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
            case 1:
                L.accept();
                break;
            case 2:
                cout << "Enter the target element to search: ";
                cin >> target;
                if (L.Search(target) != -1) {
```

```
                cout << "Element found at index " << L.Search(target) << "\n";
            } else {
                cout << "Element not found in the array.\n";
            }
            break;
        case 3:

            cout << "Element " << target << " found " << L.countRep(target) << " times in the
array.\n";
            break;
        case 4:

            L.Occurrence(target);
            break;
        case 5:
            return 0;
        default:
            cout << "Invalid choice\n";
      }
   }

   return 0;
}
```