SET
Code:

```
#include <iostream>
#include <set>
using namespace std;
int main() {
 set<int> mySet;
 int numOfElements;
 cout << "Enter the number of elements to add: ";
 cin >> numOfElements;
 for (int i = 0; i < numOfElements; i++) {
 int element;
 cin >> element;
 mySet.insert(element);
 }
 cout << "Elements in the set: ";
 for (int value : mySet) {
 cout << value << " ";
 }
 cout << endl << "Enter the element to be deleted: ";
 int elementToDelete;
 cin >> elementToDelete;
 mySet.erase(elementToDelete);S
 cout << "Set after deletion: ";
 for (int value : mySet) {
 cout << value << " ";
 }
 return 0;
```

Merge, sort

```
#include <iostream>
#include <algorithm>
#include <vector>
using namespace std;
int main() {
 int array1[] = {5, 19, 25, 24, 25};
 int array2[] = {50, 40, 2, 26, 11};
 vector<int> mergedVector(10);
 sort(array1, array1 + 5);
 sort(array2, array2 + 5);
 merge(array1, array1 + 5, array2, array2 + 5, mergedVector.begin());
 cout << "The resulting vector contains:";
 for (vector<int>::iterator it = mergedVector.begin(); it != mergedVector.end(); ++it)
 cout << ' ' << *it;
 cout << '\n';
 return 0;
}
```

Vector

```
#include <iostream>
#include <vector>
using namespace std;
int main() {
 vector<int> numbers;
 int numOfElements;
 cout << "Enter the number of elements: ";
 cin >> numOfElements;
 for (int i = 0; i < numOfElements; i++) {
 int value;
 cin >> value;
 numbers.push_back(value);
 }
 cout << "Elements in the vector: ";
 for (int i = 0; i < numOfElements; i++) {
 cout << numbers[i] << " ";
 }
 cout << endl << "Deleting the last element" << endl;
 numbers.pop_back();
 cout << "Elements in the vector after deletion: ";
 for (int i = 0; i < numbers.size(); i++) {
 cout << numbers[i] << " ";
 }
 return 0;
}
```

Stack

```
#include <iostream>
#include <stack>
using namespace std;
int main() {
 stack<int> myStack;
 myStack.push(34);
 myStack.push(45);
 myStack.push(56);
 myStack.push(22);
 myStack.push(19);
 cout << "Top element: " << myStack.top() << endl;
 myStack.pop();
 myStack.pop();
 cout << "Top element after two pops: " << myStack.top() << endl;
 return 0;
}
```