Exercise Class 11 – Theory

0.1 Dynamic Data Structures and Iterators

Pointers allow us to traverse contiguous memory blocks. However, we cannot use them for traversing containers because that would require us to know how these containers are implemented. Therefore, most containers provide iterators that encapsulate the functionality needed for traversing a data structure. A typical pattern of traversing a container would be as follows:

```cpp
#include <vector>

std::vector<int> cont = {8,3,1,4,6,9};
for (std::vector<int>::iterator it = cont.begin();
     it != cont.end();
     ++it) {
  std::cout << *it << " ";
}
```

In this way we can iterate not only via vector, but also over other data structures such as sets:

```cpp
#include <set>

std::set<int> cont = {8,3,1,4,6,9};
for (std::set<int>::iterator it = cont.begin();
     it != cont.end();
     ++it) {
  std::cout << *it << " ";
}
```