Removing Algorithms
Solutions

remove()

- Describe the remove algorithm function
  - remove() takes a value
  - Each element which has that value is moved to the back of the range
  - The elements which were behind it are moved forwards by one
  - remove() returns an iterator pointing to the first removed element
- What arguments does remove() take?
  - remove() takes an iterator range and the value to be removed
- Write a simple program which uses remove()

remove()

- Modify your program to display the number of elements before and after calling remove()
- Explain your results
  - The "removed" elements still exist in the container, but with undefined values
  - Accessing them will cause undefined behaviour
  - The size of the container object in memory has not changed, so size() gives the same result

erase()

- How can these elements be physically removed from the container?
    - To physically remove these elements, we call the container's erase() member function
    - We pass the returned iterator from remove() as the first argument
- Write a simple program that does this