

02393 C++ Programming Exercises

Week 4, September 19, 2016

To be handed in via CodeJudge, before September 26, 5pm
<https://dtu.codejudge.net/02393-f16/assignment/show/1266>

This assignment is about processing a sequence of commands related to a **bag** of numbers. Your program has to read a sequence of commands from **cin** and provide some reply to **cout**. The commands are:

add x : number x must be added to the bag. No output to be provided;

del x : if there is an element x in the bag, remove it, otherwise do nothing. No output to be provided;

qry x : if x belongs to the bag then output **T**, otherwise output **F**;

quit: stop your program.

The goal of the three exercises is always the same (reply to the input, assuming an initially empty bag) but the nature of the bag (set or multiset) and the members of the bag (integers or reals) vary. In particular,

Bag 1: Here you have to consider that the bag is a set of **int** values.

Example:

input: **add 1 add 2 add 1 del 1 qry 1 qry 2 quit**

output: **FT**

Bag 2: Here you have to consider that the bag is a multiset of **int** values. The main difference with respect to the previous exercise is that now repetitions are allowed. This means, for example, that deleting x just removes one occurrence of x from the multiset.

Example:

input: **add 1 add 2 add 1 del 1 qry 1 qry 2 quit**

output: **TT**

Bag 3 Here you have to consider that the bag is a multiset of **double** values.

Example:

input: **add 3.14 add 3.1415 del 3.14 qry 3.1415 qry 3.14 quit**

output: **TF**

You can solve this exercise using several data structures:

Arrays: Use arrays if you feel that you need to test your ability to deal with arrays. Exercises 1 and 2 should be easy, while Exercise 3 is an example showing some limitations/inconveniences of using arrays to implement bags.

Containers: We will properly introduce containers later in the course, but you can already try to use them to get a first feeling of how convenient they are with respect to array-based structures. In particular, you could consider these two classes of containers:

- <http://en.cppreference.com/w/cpp/container/set>
- <http://en.cppreference.com/w/cpp/container/multiset>

Hint: for every command of the exercise, there is a method/function that does (almost) what you need.