02393 C++ Programming Exercises

Assignment 3

To be handed in via CodeJudge — `https://dtu.codejudge.net/02393-e22/assignments`

1 Fun with bags

This assignment is about processing a sequence of commands that query and update 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: add number x to the bag. Do not produce any output;

`del` x: if x is in the bag, remove it; otherwise, do nothing. Do not produce any output;

`qry` x: if x is in the bag then output `T`, otherwise output `F`;

`quit`: end the program.

The goal of the two exercises below is always the same (reply to the input, assuming an initially empty bag) but the nature of the bag varies.

Fun with bags 1. Here you have to consider that the bag is a set of `int` values between 0 and 1000 (included).
Example:
input: `add 1 add 2 add 1 del 1 qry 1 qry 2 quit`
output: `FT`

Fun with bags 2. Here you have to consider that the bag is a *multiset* of `int` values between 0 and 1000 (included). 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`

Hints. You can solve both exercises using arrays, maybe reusing ideas from Assignment 2...

If you want, you can also use **C++ containers**. We will see containers later in the course, but you may 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`

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

2 Address book

This assignment is about processing a sequence of commands that query and update the pages of an address book. In particular, the address book contains 50 pages numbered from 1 to 50, and each page can be filled with information about a contact, consisting of the following fields:

- **First name**
- **Middle name**
- **Last name**
- **Phone number**
- **Street address**
- **House number**
- **ZIP code**
- **Region**

Your program has to read a sequence of commands from `cin` and provide some reply to `cout`. The commands are:

`add` p $firstName$ $middleName$ $lastName$ $phoneNumber$ $streetAddress$ $houseNumber$ $zipCode$ $region$: if page p exists and is empty, fill all its fields with the information being provided. Otherwise, do nothing. Do not produce any output.

`clr` p: if page p exists, erase all contact information of that page, making it empty. Otherwise, do nothing. Do not produce any output.

`qry` p: if page p exists and is not empty, then output the contents of $firstName$, $middleName$, $lastName$, $phoneNumber$, $streetAddress$, $houseNumber$, $zipCode$, $region$ separated by a comma, terminating with a newline (`endl`). Otherwise, do nothing.

`quit`: end the program.

Note: the commands above and their arguments are separated by spaces and/or newlines; inside each argument, you should use the underscore `"_"` instead of the space character `" "`.

Example. Suppose we provide the following sequence of inputs:

```
add 1 John B. Doe 12345678 Richard_Petersen_Plads 321 2800 Hovedstaden
add 2 Jane A. Wolford 98765432 Fredrik_Bajers_Vej 7 9220 Nordjylland
qry 1 qry 2
quit
```

Then, the program output should be:

```
John,B.,Doe,12345678,Richard_Petersen_Plads,321,2800,Hovedstaden
Jane,A.,Wolford,98765432,Fredrik_Bajers_Vej,7,9220,Nordjylland
```

Hints. You can solve the exercises using an array of structures to represent the address book, and an enumeration to represent the 5 regions of Denmark: Hovedstaden, Midtjylland, Nordjylland, Sjaelland, Syddanmark.. . .

If you want, you can also use **C++ containers**. We will see containers later in the course, but you may 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`

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