LAB ASSIGNMENT 3-4

Details for No stone unturned

Bower looks over the maps you've been sorting and says she's going to need help with another issue. She wants to check if you can instantly search for maps that are not to old, and at the same time delete or re-focus some map photos that have become nigh-unreadable with age.

You want to see all maps whose years of storage do not reach a given value ordered ascending by state of deterioration. The interface must be able to undo/redo any number of mistakes you do.

* All commands must be in the form:
  + list yearsOfStorage
  + undo
  + redo
* Static allocation starts to limit the old hardware, your laziness has caught up with you, time to switch to dynamic memory allocation. (define a vector structure with a dynamically allocated array)
* Write notes (specification) on your work and (unit) test everything.
* The memory being limited you have to make sure it does not leak.

**Bonus**

You hear of extra rewards for you third and forth assignment.

1. Remember to use function pointers for: (deadline: **week 4**, bonus: **0.1p**)
   * For your third assignment add a different type of filtering (your choice).
   * For your forth assignment add descending sorting.
2. For your forth assignment, provide 2 different implementations for the undo/redo functionality: one using a list of operations and oneusing the "list of lists" approach. Implementyour dynamic array generically, such that it can contain any type of elements (use void\*). Use this structure for your repository, as well as to implement the "list of lists" approach for the multiple undo/redo functionality. (deadline: **week 5**, bonus: **0.3p**)

(There are no automated tests defined for bonuses, present them manually)