Programming Exam 2: STL (Example)

This sample exam has the same format of **programming exam 2**.

The example project has a file, **Functions.h**, that contains two empty classes:

- **DArrayChild**, which **inherits** from the **DArray** class
- **DoublyListChild**, which **inherits** from the **DoublyList** class

Note that both the parent classes have **protected** member variables; therefore, they can be accessed from the child classes directly, without the use of an accessor function.

STL functions you are allowed to use:

- Any of the functions for which descriptions are provided
- Any of the functions listed below:
  - Constructors
  - begin, end, rbegin, rend
  - size, resize, reserve
  - reserve
  - empty
  - push, push_back, push_front
  - pop, pop_back, pop_front
  - at, front, back, first, second
  - overloaded subscript operator
  - overloaded assignment operator
  - overloaded comparison operators

Implement the following functions by writing your implementation in the **Functions.h** file where indicated—**NO** need to write the function declarations and **NO** need to write the class qualifier and the scope resolution on the definitions (why? Because the definitions are written in the class definition).

You can **comment/uncomment** function calls in each **TestFunction#.h** file where indicated, to test one function at a time.

- Function **function1**
  - **Member** function of the **DArrayChild** class.
  - **Parameters:** An **STL vector** of integers
    - The function returns true if the sequence of integers in the **DArray** object is the same (same order as well) as the one in the **vector** object; if not the same, it returns false.

- Function **function2**
  - **Member** function of the **DoublyListChild** class.
  - **Parameters:** An **STL list** of integers
    - The function searches the STL list to find if the list contains the value stored in the first node of the doubly-linked list. If the element is found, the function inserts the element to the end of the doubly-linked list; if it is not found, it inserts a 0.
    - Use the **STL algorithm find** to search the STL list.
  - May use **auto**.
  - **Assumptions:** Both lists have at least one element.

- Function **function3**
    - **Member** function of the **DoublyListChild** class.
    - **Parameters:** An **STL list** of integers named **list1** and another **STL list** of integers named **list2**.
        - Using the **STL list function splice**, insert the first element stored in list 2 in the third position of list 1, and then copy in reverse all the elements in list 1 into the calling object.
        - Example:

List 1:	45,87,12,35,94,21,23,14,82
List 2:	**56**,34,87,23,14,56,45
Resulting list 1:	45,87,**56**,12,35,94,21,23,14,82
Resulting list 2:	34,87,23,14,56,45 (function splice removes the element)
Resulting dll:	82,14,23,21,94,35,12,**56**,87,45

    - **Restrictions:** Do **not** use auto.
    - **Assumptions:**
        - Both lists have several elements.
        - Calling object is empty.