**4. How is a vector different from an array? (5 pts)**

Vectors and arrays are both data structures able to store multiple elements, yet they differ from each other based on their characteristics and behaviour.

1. **Resizable:**

**Array**: An array has a fixed size, and we have to define its length. It is one of the non-infinite data structures in any programming language, so we must know how many items we are going to store ahead of time.

**Vector**: Vectors are dynamic as opposed to arrays. Not only we can create a vector of an initially small size, but it also grows dynamically as more items are added. It manages memory making vectors more versatile.

1. **Memory Management:**

**Array**: arrays have a fixed size, memory allocation is done upfront. Arrays allocate a memory block, keeping all their elements stored side by side in the computer's main memory (RAM). This allows some operations, like accessing a specific element by index, to be faster.

**Vector**: Vectors, being dynamic, handle memory differently. They attempt to create contiguous memory, but when they run out of space, they allocate new memory, often doubling the size. This reallocation process makes vector insertions slightly slower compared to arrays, especially when there are large reallocations.

1. **Performance Impact:**

**Array**: Since the size is fixed, arrays provide better performance if you have a known, constant number of elements to store. Accessing an element by index is a constant-time operation (O(1)).

**Vector**: Vectors, while dynamic, come with the overhead of resizing. As the vector grows and needs to reallocate memory, the time complexity for inserting elements becomes O(n) in the worst-case scenario, although the amortized time is O(1).

1. **Use Case:**

**Array**: Arrays are the best choice when we have a fixed-size structure with predictable performance, especially if our data will always be small and predetermined. For example, arrays are often used at a lower level in programming for speed and simplicity.

**Vector**: Vectors are preferable when we need flexibility, such as when the size of the data points changes over time and we don’t know in advance how many elements will be stored.

**7. This question is not graded and only aims to gather feedback for this assignment. Did you already know (or have some experience with) C++ before this class? Do you feel that they helped you get a basic grasp of the language? Why/why not? Are there any other concepts that you think should be covered in an introduction to the language? If so, what concepts?**

Yes, I had some prior experience with C++ during my undergraduate studies. This assignment helped me to regain and solidify the concepts of C++ that I had previously learned. Revisiting these topics through the tasks presented in this assignment provided me with a clear recollection of fundamental C++ concepts, such as working with arrays, vectors, file input/output, and basic operations like the Hadamard product. All the concepts covered in this assignment have been helpful in reinforcing my understanding of C++.