Exercise#2

1)Array Size is the actual number of elements stored in the array. Capacity is the total amount of space allocated for the array, indicating how many elements it can hold without requiring resizing.

2) When an array's size exceeds its capacity, it needs to grow to accommodate more elements. This involves creating a new, larger array and copying the elements from the old array to the new one. The memory layout before expansion has the array filled to its capacity. After expansion, the new array has a larger capacity with the same elements copied over, plus additional space for more elements.

1. With Space After the Array: The expansion is straightforward if there's free memory adjacent to the array's end; the array can potentially grow into this space.
2. Without Space (Memory Occupied): If the memory following the array is occupied, a new larger array is allocated elsewhere in memory, and elements are copied over. The old array is then deallocated.

3)Real-world implementations use strategies to minimize the performance impact of array expansion. One common technique is to increase the array's capacity by a factor each time it grows. This approach spreads the cost of resizing over many insertions, making the average cost of insertion constant over time.