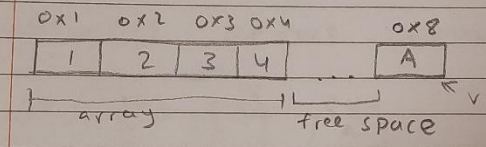
1) Array size refers to the number of elements currently stored in the array. It represents the actual data held within the array.
Array capacity refers to the maximum number of elements that the array can hold without needing to resize. It represents the total allocated memory for the array.

2)

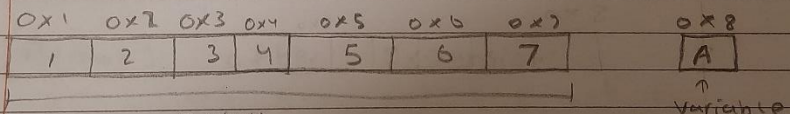
Q2) case where there is space in memory after the end of array:

- The array can simply expand into avaliable memory space. The size of array is increased and additional elements can be added to the array:

memory before expansion:



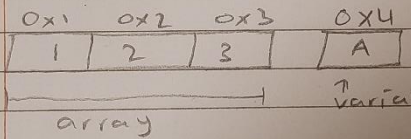
memory after expansion:



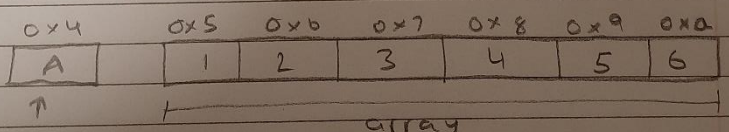
case where memory after the end of the array is occupied by another variable:

- The array cant expand in its current memory location because the memory space is not contiguous. In this case new block of memory is allocated elsewhere that is large enough to hold expanded array. Existing elements are copied from old array to new one. The old array memory is deallocated.

memory before expansion:



memory after expansion:



3) Doubling the size of the array: When the array is full, instead of increasing the size by just one element, the size of the array is doubled. This means that insertions can be done in constant amortized time. The reason is that doubling allocates spaces for future elements as well, reducing the number of times reallocation needs to happen.