ENSF 338 Lab 4 Exercise 2

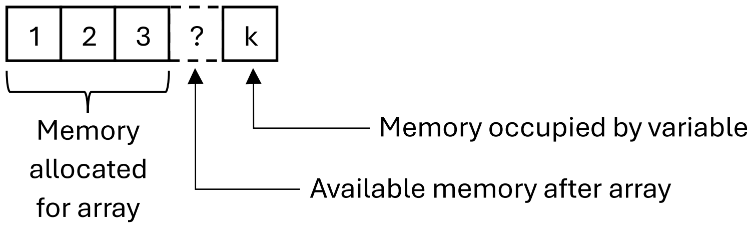
1. Explain the difference between an array size and capacity.

   Array size is the number of elements currently in the array. Array capacity is the amount of memory that has been allocated for the array. Consider the array [1, 2, 3, *null*, *null*]; this array would have a size of 3 since there are 3 elements and a capacity of 5 since it can contain up to five elements.

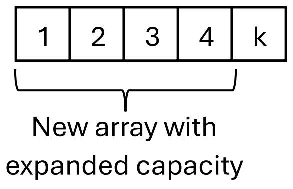
2. What happen when an array needs to grow beyond its current capacity? Explain and produce a diagram showing the memory layout before and after expansion.

   a. First, consider the case where there is space in memory after the end of the array.

      If there is space in memory, the capacity will simply be expanded and the new element will be appended to the array. Consider the array [1, 2, 3,] in a continuous block of memory; say we want to append the number 4 to the array.

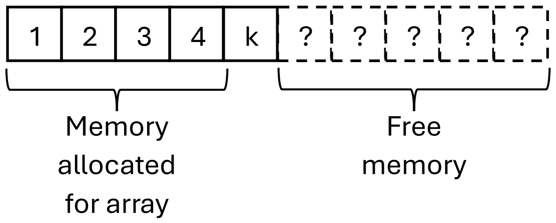

1
2
3
?
k
Memory allocated for array
Memory occupied by variable
Available memory after array

      Appending the number 4 to the array:

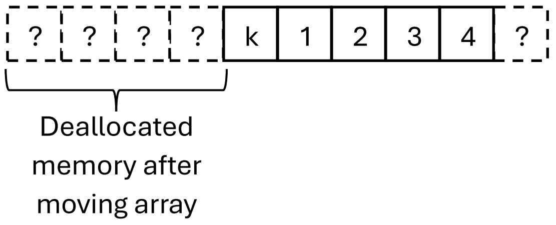


b. Then, consider the case where the memory after the end of the array is occupied by another variable.

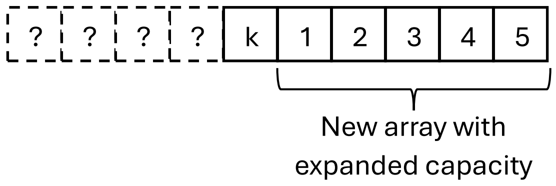
If the memory after is occupied by another variable the entire array will have to be moved (copied) to another location in memory that can accommodate the new expanded capacity. After being moved, the capacity will be expanded and the new element will be appended to the array. Consider the array [1, 2, 3, 4] from the previous example where the next memory space is occupied by a variable; say we want to append the number 5 to the array.



Copying the array to free memory:



Appending the number 5 to the array:



3. Discuss one or more techniques real-world array implementations use to amortize the cost of array expansion.

   One technique to amortize the cost of array expansion is the use of a growth factor. Different languages use different growth factors for array expansion, for example Python uses 1.125, and Java uses 2 for Vector and 1.5 for ArrayList. A growth factor determines how much to grow expand the array by (regardless of how much capacity is immediately necessary). Consider an array with a capacity of 10 that is expanded with a growth factor of 1.5. The new capacity of the expanded array would be 15 (10 x 1.5 = 15). This reduces the number of array expansions overtime since as the array gets larger, the expanded capacity also becomes larger. For example, if the array grew to a capacity of 1000, its expanded capacity would be 1500 (1000 x 1.5 = 1500). This large increase in capacity allows more elements to be appended without having to move/copy the entire array to another memory location which is costly due to its complexity of O(n).