



SIMON FRASER UNIVERSITY  
ENGAGING THE WORLD

# CMPT 125 - Introduction to Computing Science and Programming II

Lab 7. Using structs to create python-style arrays

# Arrays

- Collection of items stored at contiguous memory locations
- Idea to store multiple items of the same type together
- Each element can be uniquely identified by its index in the array
- Advantages:
  - Allows random access to elements
  - Ability to represent multiple data items of the same type using a single name

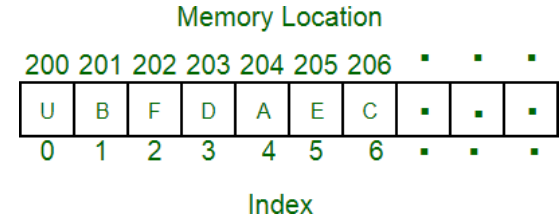


Fig1: Array Structure  
Source: [geeksforgeeks](https://www.geeksforgeeks.org/)

# Array Operations

- **Get(int index):** Get a value stored at a particular index
  - Time Complexity:  $O(1)$
- **Set(int index, int value):** Set a value at a particular index
  - Time Complexity:  $O(1)$
- **Append(int value):** Appends value to the end of the array.
  - **Best Time** Complexity:  $O(1)$
  - **Worst Time** Complexity:  $O(N)$
- **Print():** Print all values of the array
  - Time Complexity:  $O(N)$

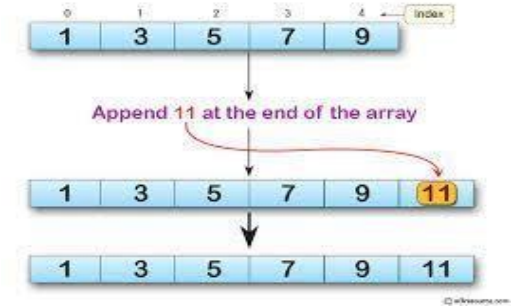


Fig2: Array Append

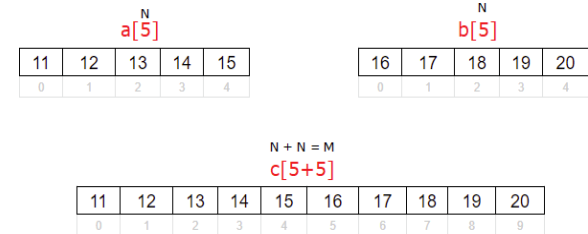


Fig3: Array Concatenation

# Array Operations

- 3 files in the zip folder:
  - `my_array/my_array.c`
  - `my_array/my_array.h`
  - `my_array/test_my_array.c`
- "my\_array" allows us to use array like in Python without worrying about resizing
- Functions implemented for operations to get, set, append and print values in an array

## Exercise

- Read and understand the functions defined in `my_array.c`
- Implement the function **`extend()`** in `my_array.c`, which concatenates the values of one array to another array
- Modify the **`append()`** function, to double the capacity of the array instead of increasing it by 1.