CPSC 331 — Solution for Question #5 on the Practice Midterm Test

This question concerned the **bounded stack** abstract data type.

(a) You were first asked to describe how an **array** (or `ArrayList`) `A` can be used to implement a **bounded stack**. You were asked where the value at the **bottom** of the stack be should stored in the array and (if this stack has size n and the array has length $m \geq n$), where the value at the **top** of the stack should be stored, if stack operations are to be implemented efficiently, and where the *other* values on the stack should be stored.

Location of Bottom of Stack: `A[0]`

Location of Top of Stack: `A[`$n-1$`]`

Location of Other Values of Stack: Positions from 1 to $n-2$, in order from oldest to newest

(b) You were next asked to give pseudocode or a description in English for a method to remove (or **pop**) a value `x` from a stack when this array-based implementation is being used — assuming that the `size` of the stack is also being stored.

Method To Pop a Value Off of a Stack — Brief Description in English: If the stack is already empty — so that its `size` is already 0 — throw an exception and do not change the array. Otherwise set `x` to be the value $A[\texttt{size} - 1]$, decrement `size` and return `x`.

Method to Pop a Value Off of a Stack — Pseudocode

```
if (size == 0) {
  throw a NoSuchException
} else {
  x = A[size-1];
  size = size − 1;
  return x;
}
```