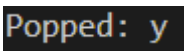
Version 1 - Basic Implementation

1. The program prints out the entire array concatenated together. This is because the program treats an array of chars as a string.
2. 

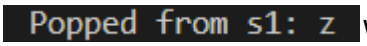
First, the program pushes “abcde” onto the stack. When the stack is popped, the index of the array decreases, however, the data itself is not removed. The pop function, on the other hand, assigns the value at the current index of the stack. So when the stack is popped four times, the index just decreases from 5 to 1. Then when the program pushes “fg” it simply replaces the indices 1 and 2.

Version 2 - Encapsulation and Modular Programming

1. 

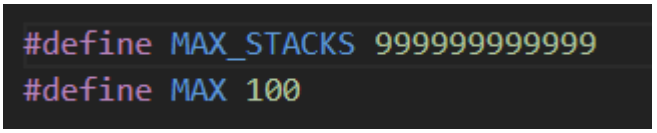
Instead of outputting “z”, the program outputs “y”. This is because assigning 2 to top moves the top of the stack to the index of “y” which then gets popped.

Version 3 - Structs with Pointers

1. 

With structs you are still able to access and modify top, and thus, modify the Store array without the use of push or pop.

Version 4 - Structs with Handlers

1. 

If the variables are improperly defined, it may lead to errors such as stack overflows.

Version 5 - OOP

1. The programmer needs to quickly implement a solution without the extra overhead
2. Directly accessing the variables is needed in the program for optimization and protecting variables is unnecessary