Lab 2.2a – Character Stack

Learning Goals

1) Develop your ability to program in C and use a Unix shell.
2) Develop your understanding of the stack data structure.

Your Task

Implement a stack in C for storing characters.

An array works well for storing a stack. The bottom of the stack is at the beginning of the array. The stack grows to the right. The position of the first empty position on the stack will be stored in the variable, sp. (Said another way, sp always points one to the right of the top of the stack.)

main() will test your implementation and has already been written for you.

Starter code, including the variable definitions and the function declarations, has been created for you.

Your stack has four behaviors (functions):

push(c):

- Pushes c onto the stack.
- Returns 0 if successful, -1 otherwise.

pop():

- Pops and returns the top element on the stack.
- Returns -1 if the stack is empty

stackSize():

- Returns the current size of the stack.

printStack():

- Prints the stack positions and contents. An empty stack should print nothing.
- See comments in starter code for the specific format.

Sample Run, passing all tests.

```
$ gcc -o charStack charStack.c

$ ./charStack
Passed 1
Passed 2
Passed 3
0: A
1: C
2: D
3: E
Passed 4
Passed 5
Passed 6
Passed 7
Passed 8
Passed 9
```