

## SA-6 Problem 1.

Siddhant Keshri

Empty stack  $s \rightarrow []$

$\hookrightarrow \text{push}("a") \rightarrow [a]$

$\hookrightarrow \text{push}("b") \rightarrow [a, b]$

$\hookrightarrow \text{pop}() \rightarrow [a]$  // returns "b"

$\hookrightarrow \text{push}("c") \rightarrow [a, c]$

$\hookrightarrow \text{pop}() \rightarrow [a]$  // returns "c"

$\hookrightarrow \text{pop}() \rightarrow []$  // returns "a"

$\rightarrow$  draw the concrete state of an SLL stack after each operation

$\hookrightarrow \text{Empty} : \text{head} \rightarrow [ ]$

$\hookrightarrow \text{push}("a") \rightarrow \text{head} \rightarrow a \rightarrow [ ]$

$\hookrightarrow \text{push}("b") \rightarrow \text{head} \rightarrow b \rightarrow a \rightarrow [ ]$

$\hookrightarrow \text{pop}() \rightarrow \text{head} \rightarrow a \rightarrow [ ]$  // returns "b"

$\hookrightarrow \text{push}("c") \rightarrow \text{head} \rightarrow c \rightarrow a \rightarrow [ ]$

$\hookrightarrow \text{pop}() \rightarrow \text{head} \rightarrow a \rightarrow [ ]$  // returns "c"

$\hookrightarrow \text{pop}() \rightarrow \text{head} \rightarrow [ ]$  // returns "a"

Problem 2.)

Empty Queue < >

↳ enzyme ("a")  $\rightarrow$  <a>

 $\hookrightarrow \text{engineer}("b") \rightarrow \langle a, b \rangle$ 

↳  $\text{degree}() \rightarrow \langle b \rangle$  returns "a"

↳ engine ("i")  $\rightarrow \langle b, c \rangle$

↳  $\text{define}() \rightarrow \langle c \rangle$  // returns "b"

↳ `degene()` → `<>` // returns ""

→ Draw SLQ type implementation:

Empty Queue : head  $\rightarrow$  [1]  
tail

L<sub>1</sub> engine ("k"): head → α → [ \ ]  
                                    ↑ tail

$\hookrightarrow$  engine ("b"): head  $\rightarrow a \rightarrow b \rightarrow [\ ]$

$\hookrightarrow$  dequeue(): head  $\rightarrow \downarrow \rightarrow [\ ]$  // returns "a"  
                                 $\uparrow$  tail

↳ engine (""): head  $\rightarrow$  b  $\rightarrow$  c  $\rightarrow$  [...]   
 tail

↳ dequeue(): head  $\rightarrow$  c  $\rightarrow$  1) // returns "5"

↳ degenerate(): head  $\rightarrow$  [1] // returns "1"  
                                   ↑ tail