



1.4.2 Stacks

Q1

a) Describe **two** differences between a stack and a queue.

[4 marks]

- Stack is LIFO
- Queue is FIFO
- Stack uses one pointer (for top)
- Queue uses two pointers (for start and end)
- Stack: data is pushed/popped from the top
- Queue: data is enqueued onto the back and dequeued from the start

b) Draw what the stack show below would look like after the following operations.

push("A"), push("B"), pop(), push("C"), pop(), push("D")

	D
	A
X	X
Y	Y
Z	Z
Before Operations	After Operations

[2 marks]





The diagram below shows a stack before and after a number of operations have been carried out.

X	
Y	A
Z	Z
State 1	State 2

c) List the operations needed to get the stack from State 1 to State 2.

[2 marks]

- `pop()`
- `pop()`
- `push("A")`

d) Complete the pseudocode function for a `pop()` operation where `items` is an array containing the data for the stack and `top` is a pointer to the item at the top of the stack.

[4 marks]

```
function pop()
    if top == -1 then
        return -1
    else
        item = items[top]
        top = top - 1
        return item
    endif
endfunction
```





Q2

A function called *reverse* uses a stack called *theStack* to reverse data that is passed in as a parameter called *name*. For example, the name “Jack” would be returned as “kcaJ” by the function.

theStack uses these operations which are already defined as global scope in the program:

- *push()* – This will add data that is passed in as a parameter to the stack
- *pop()* – This will remove and return the item on top of the stack.

Write the function *reverse* so that it:

- accepts the name as a parameter
- uses *push()* to add each character in the name to *theStack* separately
- uses *pop()* to return each character from *theStack* and add it a variable called *reverseName*
- outputs the variable *reverseName* once all the characters have been popped from *theStack*.

[5 marks]

```
function reverse(name)
    reverseName = ""
    for nameCount = 0 to name.length-1
        theStack.push(name[nameCount])
    next nameCount
    for nameCount = 0 to name.length-1
        reverseName = reverseName + theStack.pop()
    next nameCount
    return reverseName
endfunction
```

