Algorithms and Data Structures: Practical 4 (SOLUTIONS)

Question 1

(a) Queue

(b) Stack

(c) Queue

(d) Stack

Question 2

operation	output	queue
enqueue(4)		4
enqueue(6)		4 6
dequeue	4	6
enqueue(2)		6 2
dequeue	6	2
front	2	2
dequeue	2	
dequeue	error	
isEmpty	true	
enqueue(1)		1
enqueue(2)		1 2
size	2	1 2
enqueue(6)		1 2 6
enqueue(4)		1 2 6 4
dequeue	1	2 6 4

Question 3

operation	output	stack
push(2)		2
push(9)		2 9
push(8)		2 9 8
pop	8	2 9
pop	9	2
top	2	2
pop	2	
pop	error	
isEmpty	true	
push(1)		1
push(3)		1 3
push(5)		1 3 5
size	3	1 3 5
push(2)		1 3 5 2
pop	2	1 3 5

For the last three questions, pseudocode is presented. I suggest you work through these using a specific example (such as the ones given in the questions) to understand how they work.

Question 4

```
Input: postfix expression p
Output: evaluated value of p
  stack S
  for element in p do
    if element is a number then
      S.push(element)
    else
      temp1 = S.pop
      temp2 = S.pop
      newValue = temp2 element temp1          // element is one of + − ∗/
      S.push(newValue)
    end if
  end for
  result = S.pop                              // only one number left in the stack
  return result
```

Question 5

```
Input: array of n daily shares prices A
Output: array of "days since higher price" values
  stack S
  array B of size n with all cells empty
  for i = 0 to n - 1 do
    while S.isEmpty is false and A[S.top] ≤ A[i] do
      S.pop
    end while
    if S.isEmpty then
      B[i] = ∗
    else
      B[i] = i - S.top
    end if
    S.push(i)
  end for
  return B
```

Question 6

Note that $*$ and / have precedence over $+$ and $-$. Also one operator has precedence over another if it appears earlier (farther left) in the infix expression.

```
Input: infix expression f
Output: equivalent postfix expression
  stack S
  string p = ''
  for element in f do
    if element is a number then
      p = p + element
    else
      while S.isEmpty is false and S.top has precedence over element do
        p = p + S.pop
      end while
      S.push(element)
    end if
  end for
  while S.isEmpty is false do
    p = p + S.pop
  end while
  return p
```