Problem 1

a) Insert Algorithm:

```
1: function SEARCH_DOUBLY_LINKED_LIST(L, x)
2:     e ← L.head
3:     while e.key ≠ x and e ≠ NIL do
4:         e ← e.next
5:     end while
6:     return e
7: end function
```

Search Algorithm:

```
1: function INSERT_DOUBLY_LINKED_LIST(L, x)
2:     e ← new element
3:     e.key ← x
4:     e.next ← L.head
5:     L.head.prev ← e
6:     L.head ← e
7:     return L
8: end function
```

b)

```
1: function DOUBLY_LINKED_LIST_APPEND(A, B)
2:     a_i ← A.head
3:     while a_i.next ≠ NIL do
4:         a_i ← a_i.next
5:     end while
6:     ▷ To keep list B intact, it seems to be not possible to simply link the
       head of B to the end of A
7:     e ← new element
8:     b_i ← B.head
9:     a_i.next ← B.head
10:    B.head.prev ← a_i.next
11:    return A
12: end function
```

c)

```
1: function DOUBLY_LINKED_LIST_ZIP(A, B)
2:     i ← 1
3:     a_i ← A.head
4:     b_i ← B.head
5:     while i < n do
6:         ▷ Repointing a_i.next, b_i.prev, b_i.next and a_{i+1}.prev
7:         a_{i+1} ← a_i.next
8:         a_i.next ← b_i
9:         b_i.prev ← a_i
10:        b_{i+1} ← b_i.next
```

```
11:         b_i.next ← a_{i+1}
12:         a_{i+1}.prev ← b_i
13:         ▷ Initializing the variables for the next iteration
14:         a_i ← a_{i+1}
15:         b_i ← b_{i+1}
16:         i ← i + 1
17:     end while
18:     a_i.next ← b_i
19:     b_i.prev ← a_i
20:     return A
21: end function
```