Algorithm for inserting a new node in the beginning of a doubly linked list

Input: A pointer to the first node of a doubly linked list, say HEAD and an item, say ITEM to insert in the beginning of the list.

Output: ITEM inserted in the beginning of the list or suitable unsuccessful message.

Data structure used: A doubly linked list with HEAD holding the address of the first node of the list where each node contains a data field, say DATA, an address field holding the address of the immediate next node, say NEXT and an address field holding the address of the immediate previous node, say PREV.

Steps:

1. Begin
2. Set newnode = getnode()
3. If newnode = NULL
4. Then
5. Print “Memory is not available, insertion not possible”
6. Else
7. Set newnode →DATA = ITEM
8. Set newnode →PREV = NULL
9. Set newnode →NEXT = HEAD
10. If HEAD! = NULL
11. Then
12. HEAD →PREV = newnode
13. End If
14. Set HEAD = newnode
15. End If
16. End

Algorithm for inserting a new node at the end of a doubly linked list

Input: A pointer to the first node of a doubly linked list, say HEAD and an item, say ITEM to insert at the end of the list.

Output: ITEM inserted in the beginning of the list or suitable unsuccessful message.

Data structure used: A doubly linked list with HEAD holding the address of the first node of the list where each node contains a data field, say DATA, an address field holding the address of the immediate next node, say NEXT and an address field holding the address of the immediate previous node, say PREV.

Steps:

Begin

Set newnode = getnode()

If newnode = NULL

```
Then
        Print "Memory not available to insert new node"
Else
        Set newnode →DATA = ITEM
        Set newnode →NEXT = NULL
        Set temp = HEAD
        If temp != NULL
        Then
                While (temp →NEXT != NULL)
                Begin
                        Set temp = temp →NEXT
                End While
                Set temp →NEXT = newnode
        End If
        Set newnode →PREV = temp
        If HEAD = NULL
        Then
                Set HEAD = newnode
        End If
End If
End
```

Algorithm for inserting a new node at a specific position of a doubly linked list

Input: A pointer to the first node of a doubly linked list, say HEAD and an item, say ITEM to insert at a specific position, say POS of the list.

Output: ITEM inserted in the beginning of the list or suitable unsuccessful message.

Data structure used: A doubly linked list with HEAD holding the address of the first node of the list where each node contains a data field, say DATA, an address field holding the address of the immediate next node, say NEXT and an address field holding the address of the immediate previous node, say PREV.

Steps:

```
Begin
Set temp = HEAD
Set count = 0
While temp != NULL
Begin
        Set count = count + 1
```

```
        Set temp = temp →NEXT
End While
If (POS < 1 Or POS > count + 1)
Then
        Print "Invalid position specified, insertion is not possible"
Else
        Set newnode →DATA = ITEM
        Set temp = HEAD
        Set i = 1
        While (i < POS – 1)
        Begin
                Set temp = temp →NEXT
                Set i = i + 1
        End While
        If POS = 1
        Then
                Set newnode →PREV = NULL
                Set newnode →NEXT = temp
                If count != 0
                Then
                        Set temp →NEXT = newnode
                End If
                Set HEAD = newnode
        Else
                Set newnode →NEXT = temp →NEXT
                Set newnode →PREV = temp
                Set temp →NEXT = newnode
                If POS != count + 1
                Then
                        Set (newnode →NEXT) →PREV = newnode
                End If
        End If
End If
End
```

Algorithm for inserting a new node after a specific value of a doubly linked list

Input: A pointer to the first node of a doubly linked list, say HEAD and an item, say ITEM to insert after a node with a specific value, say VAL in the list.

Output: ITEM inserted after the node with VAL in the list or suitable unsuccessful message.

Data structure used: A doubly linked list with HEAD holding the address of the first node of the list where each node contains a data field, say DATA, an address field holding the address of the immediate next node, say NEXT and an address field holding the address of the immediate previous node, say PREV.

Steps:

```
Begin
Set temp = HEAD
While (temp != NULL)
Begin
        If (temp →DATA = VAL)
        Then
                Break
        End If
        Set temp = temp →NEXT
End While
If temp = NULL
Then
        Print “VAL not found in any node of the list, insertion not possible”
Else
        Set newnode →NEXT = temp →NEXT
        Set newnode →PREV = temp
        If (temp →NEXT = NULL)
        Then
                Set (newnode →NEXT )→PREV = newnode
        End If
        Set temp →NEXT = newnode
End If
End
```