

	Course:	Data Structure	Course Code:	
	Program:	BSCS	Semester:	4th
	Name:	Mukhammad Mustafa	Section:	4B
	Registration #:	L17-4116	Time:	20 mins
			Assessment	Quiz 2

Q1: Given a linked list and a value x , partition it such that all nodes less than x come before nodes greater than or equal to x .

Example:

Input: head = 1 → 4 → 3 → 2 → 5 → 2 and $x = 3$

Output: 1 → 2 → 2 → 4 → 3 → 5

- Write down an algorithm for the aforementioned problem (Use any combination of your own imagination)
- Write down code in C++ to accomplish the aforementioned task

5

a) ~~if (head == NULL)~~

Creating 2 linked lists, Lower & Greater

~~if (head == NULL)~~

Traversing through the original Linked List and storing the element in Lower if it is lesser than 3. ~~and~~ storing the element in greater if it is greater than 3. Finally merging the 2 Linked Lists.

Code:

```
void Partition(LinkedList LL, Key)
{
    if (head == NULL)
    {
        return ;
    }
    else {
        LinkedList lower, greater;
        aux = head;
        while (aux → next != NULL)
        {
            if (aux → data < Key)
            {
                lower.insertAtStart(aux → data);
            }
        }
    }
}
```

(continued on next page)

10/10/17

```
greater.insertAtStart(aux->data);
```

```
{
```

```
aux = aux->next;
```

```
}
```

```
} lower;
```

```
while (lower->aux->next != NULL)
```

```
{ lower->aux = lower->aux->next;
```

```
}
```

```
lower->insertAtEnd(key);
```

```
lower->Merge(greater);
```

// stores the key(3) in the end of first linked list.

// Merges the two linked list.

The greater linked list is linked to end of lower linked list.