# Lab04: Linked lists

# Question 1

1. List **THREE (3)** basic operations involved with linked-lists. (3 marks)

Insertion, Deletion, Search

(b) Given a list of elements as below:

“England”, “Spain”, “Brazil”, “France”, “South Africa”

* 1. Illustrate the above list when “Malaysia” and “Singapore” are added to the front of the list.

“Malaysia”, “Singapore”, “England”, “Spain”, “Brazil”, “France”, “South Africa”

* 1. Illustrate the above list from part (i) when “Spain” is deleted and “Italy” is added to the end of the list.

“Malaysia”, “Singapore”, “England”, “Brazil”, “France”, “South Africa”, “Italy”

(7 marks)

**Question 2**

1. Explain what happens when each of the following statements is executed:
2. while (Head->Link != NULL)

Head = Head->Link;

At the end of execution, the Head pointer will point towards the last node in the linked list. This is because the pointer is assigned with its successive link member that point towards the next node until it points to the last node, of which its link member points to NULL.

(4 marks)

1. delete Head;

The first node of the linked list will be deleted and the linked list nodes becomes inaccessible unless a new pointer is created and pointed to the new first node

(2 marks)

**Question 3**

Given the following structure, answer the questions below:

struct nodeType {

char info;

node \*link;

};

nodeType \*head, \*A, \*B, \*node1, \*node2;

head

L

D

S

A

T

A

B

1. What is the output of each of the following C++ statement?
2. cout << head->info; T
3. cout << A->info; A
4. cout << B->link->info; L
5. cout << head->link->link->info; S
6. Write C++ statements to do the following.
7. Make A point to the node containing info ‘S’.

A = A -> link;

1. Make head point to the node containing ‘A’.

head = A; OR IF ‘A’ POINTER HAVE CHANGED PLACE head = head->link;

1. Make B point to the last node in the list.

B = B -> link;

1. Make head point to an empty list.

head = NULL;

1. Write C++ statements to perform the following tasks:
2. Create 2 new nodes and assigns to a pointer named node1 and node2, respectively.

nodeType newNode1 = new nodeType;

nodeType newNode2 = new nodeType;

node1 = &newNode1;

node2 = &newNode2;

1. Assign a value ‘G’ to the node1, and a value ‘K’ to the node2.

node1 -> info = ‘G’;

node2 -> info = ‘K’;

1. Insert the node1 at the front of the list.

ASSUMING head is still pointing at the first node

node1->link = head;

head = node1;

1. Insert the node2 in between node ‘A’ and ‘S’.

ASSUMING ‘A’ pointer points to node ‘S’ because of previous code:

node2->link = A;

head->link = node2;