PES University
Department of Computer Science and Engineering

UE19CS202- Data Structures and its Applications(4-0-0-4-4)
UNIT - 1 : Question Bank

1. Consider the file node.h has the following declarations; use this to answer all the sub questions in question no. 1.

```
typedef struct node{
char c;
struct node * next;
} Node;
Node * p, * q, * r;
Node x, y, z;
```

i) **For each of the following statements, either describe its effect, or state why it is illegal.**

(a) p = (Node *) malloc(sizeof(Node));
(b) * q = (Node *) malloc(sizeof(Node));
(c) x = (Node *) malloc(sizeof(Node));
(d) p = r;
(e) q = y;
(f) r = NULL;
(g) z = * p;
(h) p = * x;
(i) free(y);
(j) free(* p);
(k) free(r);
(l) * q = NULL;
(m) * p = * x;
(n) z = NULL;

ii) **Write a C function to interchange pointers p and q, so that after the function is performed, p will point to the node to which q formerly pointed, and vice versa.**

iii) **Write a C function to interchange the values in the dynamic variables to which p and q point, so that after the function is performed * p will have the value formerly in * q and vice versa.**

iv) **Write a C function that makes p point to the same node to which q points, and frees the item to which p formerly pointed.**

2. What are the advantages and disadvantages of representing a group of items as an array versus a linear linked list?
3. Write C functions to perform the following operations using Singly Linked List
    a. Append an element to the end of the list
    b. Concatenate two lists
    c. Free all the nodes in a list
4. Write C functions to perform the following operations using Singly Linked List
    a. Reverse the list, so that the last element becomes the first, and so on
    b. Delete the last element from a list
    c. Delete the n^{th} element from a list
5. Write C functions to perform the following operations using Singly Linked List
    a. Combine two ordered lists into a single ordered list
    b. Form a list containing the union of the elements of two lists
    c. Form a list containing the intersection of the elements of two lists
6. Write C functions to perform the following operations using Singly Linked List
    a. Insert an element after the n^{th} element of a list
    b. Delete every second element from a list
    c. Place the elements of a list in increasing order
7. Write C functions to perform the following operations using Singly Linked List
    a. Return the sum of integers in a list
    b. Return the number of elements in a list
    c. Move *node(p)* forward *n* positions in a list
    d. Make a second copy of a list

8. Write C functions to perform each of the operations in Question no. 3,4,5,6,7 on doubly linked lists
9. What is the average number of nodes accessed in searching for a particular element in an unordered list? In an ordered list? In an unordered array? In an ordered array?
10. Write C functions to perform each of the operations in Question no. 3,4,5,6,7 (listed above), assuming that each list contains a header node containing the number of elements in the list. (singly linked list and doubly linked list)
11. Write a C function that returns a pointer to a node containing element x in a list with a header node. The info field of the header should contain the pointer that traverses the list.
12. Write a C function to interchange the m^{th} and n^{th} elements of a list. (Singly linked list and doubly linked list)
13. Write a C function inssub(l1,i1,l2,i2,len) to insert the elements of list l2 beginning at the $i2^{th}$ element and continuing for len elements into the list l1 beginning at

position i1. No elements of the list l1 are to be removed or replaced. If i1>length(l1) + 1 (where length(l1) denotes the number of nodes in the list l1), or if i2+len-1>length(l2), or if i1<1, or if i2<1, print an error message. The list l2 should remain unchanged.

14. Write a C function search(l,x) that accepts a pointer l to a list of integers and an integer x and returns a pointer to a node containing x, if it exists, and the null pointer otherwise. Write another function, srchinsrt(l,x), that adds x to l if it is not found and always returns a pointer to a node containing x.
15. Write C functions to perform each of the operations in Question no. 3,4,5,6,7 (listed above), for circular lists. Which are more efficient on circular lists than on linear lists? Which are less efficient?
16. Write a C function multint(p,q) to multiply two long positive integers represented by singly linked circular lists.
17. Write a C function addsame to add two long integers of the same sign represented by doubly linked lists.
18. Write a C function multint(p,q) to multiply two long positive integers represented by doubly linked circular lists.
19. Explain how polynomial arithmetic of single variable can be implemented using singly linked list. (i) Specify structure of each node of the list.(ii) Show with an example of how polynomial addition and multiplication operations can be performed using linked list.
20. Specify the node structure of an integer doubly-linked list (dll). Write a function to delete a node based on the specified index position in the dll. (0 should delete the head node, 1 should delete node after the head and so on). Make sure you handle the boundary conditions.