**ASSIGNMENT 4 | 7th January 2021 DSA**

**Question 1**

**Write a function “insert\_any()” for inserting a node at any given position of the linked list. Assume position starts at 0.**

insert\_any(int a, node \*head)

{

node\* temp,p,q;

int pos;

temp=create\_node(a);

printf("Enter the position where node is to be inserted");

scanf("%d", &pos);

p=head;

q=head;

while(p->next != pos)

{

q=p;

p=p->next;

}

q->next = temp;

temp->next = p;

return (head);

}

**Question 2**

**Write a function “delete\_beg()” for deleting a node from the beginning of the linked list.**

node\* delete\_beg(node\* head)

{

node \*ptr;

ptr = head;

head = head->next;

ptr->next = null;

free(ptr);

}

**Question 3**

**Write a function “delete\_end()” for deleting a node from the end of the linked list.**

node\* delete\_end(node\* head)

{

node \*p, \*q;

p=q=head;

while(p->next != null)

{

q=p;

p=p->next;

}

q->next = null;

free(p);

}

**Question 4**

**In the Binary Search algorithm, it is suggested to calculate the mid as beg + (end - beg) / 2 instead of (beg + end) / 2. Why is it so?**

Reason:

Pointer addition is not supported in C while pointer subtraction is supported, the reason being the result of subtraction is the difference (in array elements) between the operands.

**Question 5**

**Write the algorithm/function for Ternary Search.**

Begin

if start <= end then

midFirst := start + (end - start) /3

midSecond := midFirst + (end - start) / 3

if array[midFirst] = key then

return midFirst

if array[midSecond] = key then

return midSecond

if key < array[midFirst] then

call ternarySearch(array, start, midFirst-1, key)

if key > array[midSecond] then

call ternarySearch(array, midFirst+1, end, key)

else

call ternarySearch(array, midFirst+1, midSecond-1, key)

else

return invalid location

End