1.1.2024
MONDAY

CSA0319 - DATA STRUCTURES
SSSP ALGORITHM

ASSINGMENT-03

BY

A. PAVENDHAN
192321109

1. Delete a node at the position 4 of the double linked list. Implement it

Code :-

```
#include<stdio.h>
#include <stdlib.h>
struct Node{
        int data;
        struct Node *prev;
        struct Node * next;
};
void deletenode (struct Node **head-ref, int position)
 If (*head-ref ==NULL){
    return;
 }
struct Node* temp =*head-ref;
If (Position ==1) {
   *head-ref = temp->next;
   If (*head-ref !=NULL) {
        (*head-ref)->prev =Null;
   }
   free (temp);
   return;
 }
 for (int i=1; temp !=Null && i< position; i++) {
   temp = temp->next;
 }
```

```
    if(temp ==Null) {
       return ;
    }
   If (temp -> prev ! =Null) {
      temp ->prev->next = temp ->next;
   }
   If (temp ->next ! =Null) {
      temp ->next->prev = temp -> prev;
   }
   free (temp);
 }
 void printlist(struct Node *node){
    while (node != Null) {
       printf ("%d<=> ", node ->data);
        node = node -> next;
     }
     printf("\n");
   }
int main () {
      Struck Node* head = NuLL;
      struck Node * second = NuLL;
      struck Node * third = NuLL;
      struck Node *fourth = Null;
      struck Node * fifth = Null;
      struck Node * sixth = Null;
```

```
head = (struct Node*) malloc (sizeof(struct Node));
second = (struct Node*)malloc (size of(struct Node));
third = (struct Node*) malloc (size of(struct Node));
fourth = (struct Node*) malloc (size of (struct Node));
fifth = (struct Node*) malloc (size of (struct Node));
sixth = (struct Node*) malloc (size of(struct Node));

head -> data = 22;
head -> prev = Null;
head -> next = second;

second -> data = 77;
second -> prev = head;
Second -> next = third;

third -> data = 99;
third -> prev = second;
third -> next = fourth;

fourth -> data = 22:
fourth -> prev = third:
fourth -> next = fiveth;

fifth -> data = 55;
fifth -> prev = fourth;
fifth -> next = sixth;

sixth -> data = 27
Sixth -> prev = fifth
Sixth -> next = Null;
```

```
printf(" Input: ");
printlist (head);
deletenode (&head, 4);
printf("output : ");
printlist (head);
return 0;
}
```

Input :
22<=>77<=>99<=>22<=>55<=>27<=>

22

output :
~~77~~<=>99<=>22 <=>55<=>27<=>

2: find the difference between Max and Mini elements in the S&L Input 56->99->66->22->33->.

code :

```
#include <stdio.h>
#include <stdlib.h>
struct Node {
    Int data;
    struct Node* next;
};
```

```
Int finddifference (struct Node* head) {
    If ( head == Null) {
        Printf("SLL is empty!\n");
        return 0;
    }
    Int max = head -> data ;
    int min = head -> data ;
    struct Node* current = head;
    while (current != Null ) {
        if (current -> data -> max) {
            max = current -> data ;
        }
        If (current -> data < min) {
            min = current -> data ;
        }
        Current = current -> next ;
    }
    return max - min ;
}
struct Node * createNode (int data) {
    struct Node* Newnode = (struct Node*) mallo
                             (sizeof(struct Node ));
    newnode -> data = data ;
    newnode -> next = Null;
    return newnode;
```

```
void insertnode(struct Node** head, int data) {
    struct Node *newnode = createNode(data);
    If (*head == Null) {
        *head = newnode;
    } else {
        struct Node* current = *heade;
        while (current -> next != Null) {
            current = current -> next;
        }
        current -> next = newnode;
    }
}

void displaylist (struct Node *head) {
    If (head == Null) {
        printf ("SLL is empty.\n");
        return;
    }

    struct Node * current = head;
    while (current != Null) {
        printf ("%d ->", current -> data);
        current = current -> next;
    }
    printf ("Null \n");
}

int main(){

        struct Node * head = Null;

        insertnode (&head, 56);
        insert node (&head, 99);
        insert node (&head, 66);
        insert node (&head, 22);
        insert node (&head, 33);

    printf ("Input : ");
    display list (head);
    Int difference = finddifferenee (head);
    printf ("output : Differenenee = %d \n", difference);
    return 0;

}
```

<u>output</u> :

```
Input : 56 -> 99 -> 66 -> 22 -> 33 -> Null

output : Difference = 77
```