

## Double linked List Implementation

a) Struct node {

Struct node \* prev;

int data;

Struct node \* next;

}

b) void insert() {

int i, position, new\_data;

Struct node \* ptr, \* temp;

if (head == NULL) {

printf("List is Empty");

else {

temp = head;

i = 1;

while (i < position - 1 & temp != NULL) {

temp = temp -> next;

i++;

}

if (position == 1) {

ptr -> data = new\_data;

ptr -> next = head;

ptr -> prev = NULL;

head = ptr;

}

else if (temp == NULL) {

ptr -> data = new\_data;



```

ptr → next = NULL;
ptr → prev = last;
last → next = ptr;
last = ptr;

```

```

}

```

```

else if (temp != NULL) {
    ptr → data = new_data;
    ptr → next = temp → next;
    ptr → prev = temp;
    if (temp → next != NULL) {
        temp → next → prev = ptr;
        temp → next = ptr;
    }
}
else {
    printf("Invalid position");
}
}

```

```

}

```

```

}

```

```

void Create_List() {
    struct node * ptr;
    int i, n, new_data;
    printf("Enter the no of nodes");
    scanf("%d", &n);
    if (n > 1) {
        head = (struct node *) malloc (sizeof (struct node));
        if (head != NULL) {
            printf("Enter value of Node 1:");
            scanf("%d", &new_data);
            head → data = new_data;

```



Experiment No.

Name of the Experiment

Date.

Page No.

head → prev = NULL;

head → next = NULL;

head → last = head;

for(i = 1; i ≤ n; i++)

{ ptr = (struct node \*) malloc (sizeof (struct node));

if (ptr != NULL)

{ printf ("Enter value for Node %d", i);

scanf ("%d", &new\_data);

ptr → data = new\_data;

ptr → prev = last;

ptr → next = NULL;

last → next = ptr;

last = ptr;

} printf ("Linked list created");

else

printf ("Nodes cannot be created");

}

}



```

(c) void delete() {
    struct node *ptr; struct node *temp;
    int val;
    printf("Enter the value: ");
    scanf("%d", &val);
    temp = head;
    while (temp->data != val)
        temp = temp->next;
    if (temp->next == NULL)
        printf("cannot be deleted");
    else if (temp->next->next == NULL)
        temp->next = NULL;
        printf("Last Node Deleted");
    else {
        ptr = temp->next;
        temp->next = ptr->next;
        ptr->next->prev = temp;
        free(ptr);
        printf("Node Deleted");
    }
}

```

```

(d) void display() {
    struct node *ptr = head;
    if (ptr == head) printf("Empty list");
    else {
        printf("\n\n List -> ");
        while (ptr != NULL) {
            printf("%d ", ptr->data);
            ptr = ptr->next;
        }
    }
}

```