

Name: Ramya Ramesh

USN: 1BM19CU038

Lab 8: Doubly Linked List Implementation
of Insertion of Node at the beginning and deletion
of node at specific location.

```
struct node
{
    struct node * prev;
    struct node * next;
    int data;
};
struct node * head;
```

```
void insertion_beginning()
```

```
{
    struct node * ptr;
    int item;
    ptr = (struct node *) malloc (size of (struct node));
    if (ptr == NULL)
    {
        printf ("In Overflow!");
    }
    else
    {
        printf ("In Enter the value to be inserted: ");
        scanf ("%d", &item);
        if (head == NULL)
        {
            ptr -> next = NULL;
            ptr -> prev = NULL;
            ptr -> data = item;
            head = ptr;
        }
    }
}
```

else

{

ptr → data = item;

ptr → prev = NULL;

ptr → next = head;

head → prev = ptr;

head = ptr;

}

printf("\n Node inserted");

}

void deletion_specified()

{

struct node * ptr, * temp;

int val;

printf("\n Enter the value after which node is
to be deleted : \n");

scanf("%d", &val);

ptr = head;

while (ptr → data != val)

ptr = ptr → next;

if (ptr → next == NULL)

{

printf("\n Can't be deleted!");

}

else if (ptr → next → next == NULL) {

ptr → next = NULL;

printf("\n Node deleted");

}

else {

temp = ptr → next

ptr → next = temp → next;

temp → next → prev = ptr;

free(temp)

printf("\n Node deleted!");

}

}

```
void display()
```

```
{
```

```
    struct node * ptr;
```

```
    printf("The values in the doubly linked  
list are : \n");
```

```
    ptr = head;
```

```
    while (ptr != NULL)
```

```
{
```

```
    printf("%d\t", ptr->data);
```

```
    ptr = ptr->next;
```

```
}
```

```
}
```