

LAB PROGRAM - 6.

```
#include <stdio.h>
#include <stdlib.h>

void create();
void delete(int);
void display();
void delete-first();
void delete-last();
struct node
{
    int id;
    char name[20];
    int km;
    struct node *next;
};
struct node *head = NULL;
int count = 0;

int main()
{
    int choice, ele;
    do
    {
        printf("\n1. Create \n2. Delete First Node \n\n3. Delete Nth Node \n5. Display \n\n6. Exit \n4. Delete Last Node \n");
        printf("\nEnter your choice : ");
        scanf("%d", &choice);
        switch(choice)
        {
            case 1: create(); break;
            case 2: delete-first(); break;
```

```

        case 3: printf("Enter the ID to be deleted:");
                scanf("%d", &ele);
                delete(ele);
                break;
        case 4: delete-but(); break;
        case 5: display(); break;
        case 6: printf("\nEXITING.....\n");
                break;
    }
    while(choic != 6);
    return 0;
}

```

```

void create()
{

```

```

    struct node *newnode, *temp;
    int stu_id;
    char stu_name[20];
    int semester;
    newnode = (struct node*) malloc(sizeof(struct node));
    printf("Enter the Student ID:");
    scanf("%d", &stu_id);
    printf("Enter the Student NAME:");
    scanf("%s", &stu_name);
    printf("Enter the semester:");
    scanf("%d", &semester);
    newnode->id = stu_id;
    strcpy(newnode->name, stu_name);
    newnode->sem = semester;
    if (head == NULL)
    {
        newnode->next = NULL;
        head = newnode;
        printf("Node created\n");
    }
}

```


clr
{

```
temp = head;
while (temp -> next != NULL)
    temp = temp -> next;
temp -> next = newnode;
newnode -> next = NULL;
printf("Node created \n");
```

}

void display()
{

```
struct node *ptc = NULL;
ptc = head;
if (ptc == NULL)
    printf("Empty list \n");
else
{
```

```
    printf("\n (contents of linked list)");
    printf("\n ID \n Name \n semester \n");
    while (ptc != NULL)
```

```
{
    printf("%d\t%s\t%d\n", ptc -> id,
           ptc -> name, ptc -> sem);
    ptc = ptc -> next;
```

}

}

void delete(int ele)

{

```
struct node *temp, *del = NULL;
if (head == NULL)
{
    printf("Empty list \n");
    return;
```

}

```

temp = head;
if (head->id == del)
{
    head = head->next;
    printf("Successfully deleted\n");
    return;
}

```

```

while (temp->next != NULL)
{

```

```

    if (temp->next != NULL)
    {

```

```

        del = temp->next;

```

```

        if (del->next == NULL)

```

```

        {
            temp->next = NULL;

```

```

        }

```

```

        else
            temp->next = del->next;

```

```

    }

```

```

    else

```

```

        temp = temp->next;

```

```

    }

```

```

    if (del == NULL)
    {

```

```

        printf("Student ID not found\n");
        return;
    }

```

```

}

```

```

printf("Successfully deleted\n");
}

```

```

void delete-first()
{

```

```

    struct node * todelete;

```

```

    if (head == NULL)

```

```

    {
        printf("Empty list\n");

```

```

    }

```

```

}

```



```
todelete = head;  
head = head -> next;  
printf("ID deleted = %d\n", todelete -> id);  
free(todelete);  
printf("Successfully deleted\n");  
}  
}
```

void delete-last()

```
{  
    struct node *todelete, *secondlastnode;  
    if (head == NULL)  
        printf("Empty list\n");  
    else  
    {  
        todelete = head;  
        secondlastnode = head;  
        while (todelete -> next != NULL)  
        {  
            secondlastnode = todelete;  
            todelete = todelete -> next;  
        }  
        if (todelete == head)  
            head = NULL;  
        else  
            secondlastnode -> next = NULL;  
        free(todelete);  
        printf("Successfully deleted\n");  
    }  
}
```