

LAB PROGRAM - 5.

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

void create();
void display();
void delete(int);
void insertbefore();
void insert_at_nth();

struct node
```

```
{
    int id;
    char name[20];
    int sem;
    struct node *next;
```

```
};
struct node *head = NULL;
int count = 0;
```

```
int main()
{
```

```
    int choice, eli;
    do
```

```
    {
        printf("\n 1. CREATE \n 2. INSERT at front position \n 3. INSERT at Nth position \n 4. DISPLAY \n 5. EXIT \n");
```

```
        printf("Enter your choice : \n");
```

```
        scanf("%d", &choice);
```

```
        switch(choice)
        {
```

```
case 1: create();  
        break;  
case 2: insert_before(); break;  
case 3: printf("Enter the position to insert\n");  
        scanf("%d", &ele);  
        insert_at_nth(ele); break;  
case 4: display(); break;  
case 5: printf("EXITING...\n"); break;  
}  
while(choice != 5);  
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 Student ID: \n");  
    scanf("%d", &stu_id);  
    printf("Enter the student NAME: \n");  
    scanf("%s", &stu_name);  
    printf("Enter the student SEMESTER: \n");  
    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");  
    }
```



```
else  
{
```

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

```
void display()  
{
```

```
struct node *ptr = NULL;  
ptr = head;  
if (ptr == NULL)  
printf("Nothing to Print\n");  
else  
{
```

```
printf("Contents of Linked List:\n");  
printf("\n ID \t NAME \t Semester\n");  
while (ptr != NULL)  
{
```

```
printf("%d \t %d \t %d\n", ptr->id,  
ptr->name, ptr->sem);
```

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

```
void insert-before()  
{
```

```
struct node *newnode;  
int stu-id, semester;  
char stu-name[20];
```

```
newnode = (struct node*) malloc (sizeof (struct node));  
printf ("Enter Student ID : \n");  
scanf ("%d", &stu-id);  
printf ("Enter Student NAME : \n");  
scanf ("%s", &stu-name);  
printf ("Enter the Student SEMESTER : \n");  
scanf ("%d", &semester);  
newnode->id = stu-id;  
strcpy (newnode->name, stu-name);  
newnode->sem = semester;  
newnode->next = head;  
head = newnode;  
}
```

```
void insert_at_nth ()  
{
```

```
    struct node *newnode;  
    int stu-id, semester;  
    char stu-name[20];  
    if (count+1 < p)  
        printf ("The position exceeds the number of  
                nodes \n");
```

```
    else  
    {
```

```
        printf ("Enter the Student ID : \n");  
        scanf ("%d", &stu-id);  
        printf ("Enter Student NAME : \n");  
        scanf ("%s", &stu-name);  
        printf ("Enter Student SEMESTER : \n");  
        scanf ("%d", &semester);  
newnode->id = stu-id;  
        newnode = (struct node*) malloc (sizeof (struct node));  
        newnode->id = stu-id;  
        strcpy (newnode->name, stu-name);  
        newnode->sem = semester;
```



```
if (head == NULL && p > 1)
    printf("List empty \n");
```

```
if (p == 1)
```

```
    printf("Inserted at the beginning \n");
```

```
    newnode->next = head;
```

```
    head = newnode;
```

```
    count++;
```

```
else
```

```
{
```

```
    struct node *temp1;
```

```
    temp1 = head;
```

```
    for (int i = 2; i < p; i++)
```

```
        temp1 = temp1->next;
```

```
    temp1->next = newnode;
```

```
    printf("Node Inserted at %d position
```

```
           of the list \n", p);
```

```
    count++;
```

```
}
```

```
}
```