LAB PROGRAM 5A:

```
#include <stdio.h>
#include<stdlib.h>
typedef struct Node {  int
data;  struct Node *next;
}Node;
void InsertAtBeginning( Node **head_ref,int new_data);  void
DeleteAtBeginning( Node **head_ref);  void DeleteAtEnd(
Node **head_ref);  void Delete( Node **prev_node,int pos);
void PrintList(Node * next);
void InsertAtBeginning( Node **head_ref,int new_data)
{
        struct Node* new_node=(struct Node*)malloc(sizeof( Node));
        new_node->data=new_data;  new_node->next=*head_ref;
        *head_ref=new_node;
}
void DeleteAtBeginning( Node **head_ref)
{
        Node *ptr;  if(head_ref ==
        NULL)
        {
        prin]("\nList is empty");
        }
        else
        {
        ptr = *head_ref;
        *head_ref = ptr->next;
        free(ptr);
        prin]("\n Node deleted from the beginning ...");
        }
}
void DeleteAtEnd(Node **head_ref)
{
        Node *ptr,*ptr1;
        if(*head_ref == NULL)
        {
        prin]("\nlist is empty");
```

```
        }else if((*head_ref)-> next == NULL)
        {
        free(*head_ref);  *head_ref=
        NULL;
        prin]("\nOnly node of the list deleted ...");
        }
        else
        {
        ptr = *head_ref;
        while(ptr->next != NULL)
        {
        ptr1 = ptr;
        ptr = ptr ->next;
        }
        ptr1->next = NULL;  free(ptr);
        prin]("\n Deleted Node from the last ...");
        }
}
void Delete(Node **head_ref, int pos)
{
        Node *temp = *head_ref, *prev;
        if (temp == NULL)
        {
        prin]("\nList is empty");  return;
        }
        if (pos == 1)
        {
        *head_ref = temp->next;  free(temp);
        prin]("\nDeleted node with posi,on %d", pos);  return;
        }
        for (int i = 0; temp != NULL && i < pos - 1; i++)
        {prev = temp;
        temp = temp->next;
        }
        if (temp == NULL)
        {
        prin]("\nPosi,on out of range");
```

```
        return;
        }
        prev->next = temp->next;  free(temp);
        prin](\nDeleted node with posi,on %d", pos);
        }
        void PrintList(Node *node)
        {
        while (node!=NULL)
        {
        prin](%d\n",node->data);
        node=node->next;
        }
}
int main()
{
        int ch,new,pos;  Node* head=NULL;
        prin](Santosh B\n");
        prin](1BM22CS243\n\n\n");
        while(ch!=6)
        {
        prin](Menu\n");  prin](1.Create a
        linked list\n");  prin](2.Delete at
        beginning\n");  prin](3.Delete at a
        specific posi,on\n");  prin](4..Delete at
        end\n");  prin](5..Display linked
        list\n");  prin](6..Exit\n");  prin](Enter
        your choice\n");  scanf(%d",&ch);
        switch(ch)
        {
        case 1:
        {
        prin](Enter the data you want to insert at beginning\n");
        scanf(%d",&new);  InsertAtBeginning(&head,new);
        break;
        }
        case 2:
        {
```

```
        DeleteAtBeginning(&head);
        break;
        }
        case 3:
        {prin]("Enter the posi,on at which you want to delete \n");
        scanf("%d",&pos);  Delete(&head,pos);
        break;
        }
        case 4:
        {
        DeleteAtEnd(&head);
        break;
        }
        case 5:
        {
        prin]("Created linked list is:\n");  PrintList(head);
        break;
        }
        case 6:
        {
        return 0;  break;
        }
        default:
        {
        prin]("Invalid data!");
        break;
        }
        }
        }
        return 0;
}
```

OUTPUT:

```
 Node deleted from the beginning ...Menu
1.Create a linked list
2.Delete at beginning
3.Delete at a specific position
4..Delete at end
5..Display linked list
6..Exit
Enter your choice
5
Created linked list is:
4
3
2
1
Menu
1.Create a linked list
2.Delete at beginning
3.Delete at a specific position
4..Delete at end
5..Display linked list
6..Exit
Enter your choice
4

 Deleted Node from the last ...Menu
1.Create a linked list
2.Delete at beginning
3.Delete at a specific position
4..Delete at end
5..Display linked list
6..Exit
Enter your choice
5
Created linked list is:
4
3
2
```

```
Menu
1.Create a linked list
2.Delete at beginning
3.Delete at a specific position
4..Delete at end
5..Display linked list
6..Exit
Enter your choice
3
Enter the position at which you want to delete
3

Deleted node with position 3Menu
1.Create a linked list
2.Delete at beginning
3.Delete at a specific position
4..Delete at end
5..Display linked list
6..Exit
Enter your choice
5
Created linked list is:
4
3
Menu
1.Create a linked list
2.Delete at beginning
3.Delete at a specific position
4..Delete at end
5..Display linked list
6..Exit
```