

Code:

```
#include <stdio.h>
#include <stdlib.h>
//Code by Arya Nair
struct Node
{
    int val;
    struct Node* next;
};
struct Node* head;

int isEmpty(struct Node* temp)
{
    if(temp==NULL) return 1;
    else return 0;
}

void createSSL()
{
    head=(struct Node*)malloc(sizeof(struct Node));
    head->next=NULL;
}

void insertBegin(int data)
{
    struct Node* newNode=(struct Node*)malloc(sizeof(struct Node));
    newNode->val=data;
    newNode->next=head;
    head=newNode;
}

void insertAfter(int data,int target)
{
    struct Node* temp=head;
    struct Node* newNode=(struct Node*)malloc(sizeof(struct Node));
    newNode->val=data;
    while(!isEmpty(temp)){
        if(temp->val==target){
            struct Node* temp2=temp->next;
            newNode->next=temp2;
            temp->next=newNode;
            return;
        }
        temp=temp->next;
    }
}
```

```

    }
    printf("Element not found");
    return;
}
struct Node* deleteBefore(int target)
{
    if(isEmpty(head)) return NULL;
    struct Node* temp1=head;
    if(head->val==target){
        head=NULL;
        return temp1;
    }
    if(head->next->val==target){
        head=head->next;
        return temp1;
    }

    struct Node* temp2=head->next->next;
    struct Node* temp3=head->next;
    while(!isEmpty(temp2)){
        if(temp2->val==target){
            temp1->next=temp2;
            return temp3;
        }
    }
    printf("Element not found");
    return NULL;
};

void display(struct Node* temp)
{
    while (!isEmpty(temp)){
        printf(" %d ",temp->val);
        temp=temp->next;
    }
    printf("\n");
}

int main()
{
    int data,key;
    int choice=0;

```

```

while(choice!=6){
    printf("1.Create a Linked List\n2.Insert at Beginning\n3.Insert
After a key\n4.Delete Before a Key\n5.Display the Linked
List\n6.Exit\nEnter a choice: ");
    scanf("%d",&choice);
    switch(choice)
    {
        case 1:
            createSSL();
            printf("enter a value for head Node: ");
            scanf("%d",&head->val);
            break;
        case 2:

            printf("Enter value you want to enter: ");
            scanf("%d",&data);
            insertBegin(data);
            break;
        case 3:

            printf("Enter value you want to enter: ");
            scanf("%d",&data);
            printf("Enter key: ");
            scanf("%d",&key);
            insertAfter(data,key);
            break;
        case 4:

            printf("Enter key: ");
            scanf("%d",&key);
            struct Node* deleted=deleteBefore(key);
            printf("Deleted value-%d\n",deleted->val);
            free(deleted);
            break;
        case 5:
            display(head);
            break;
    }
}

return 0;
}

```

Output:

```
Select C:\Users\Exam\Desktop\SLL.exe
1.Create a Linked List
2.Insert at Beginning
3.Insert After a key
4.Delete Before a Key
5.Display the Linked List
6.Exit
Enter a choice: 1
enter a value for head Node: 75
1.Create a Linked List
2.Insert at Beginning
3.Insert After a key
4.Delete Before a Key
5.Display the Linked List
6.Exit
Enter a choice: 2
Enter value you want to enter: 25
1.Create a Linked List
2.Insert at Beginning
3.Insert After a key
4.Delete Before a Key
5.Display the Linked List
6.Exit
Enter a choice: 3
Enter value you want to enter: 75
Enter key: 25
1.Create a Linked List
2.Insert at Beginning
3.Insert After a key
4.Delete Before a Key
5.Display the Linked List
6.Exit
Enter a choice: 5
25 75 75
```

C:\Users\Exam\Desktop\SLL.exe

```
1.Create a Linked List
2.Insert at Beginning
3.Insert After a key
4.Delete Before a Key
5.Display the Linked List
6.Exit
Enter a choice: 1
enter a value for head Node: 25
1.Create a Linked List
2.Insert at Beginning
3.Insert After a key
4.Delete Before a Key
5.Display the Linked List
6.Exit
Enter a choice: 2
Enter value you want to enter: 75
1.Create a Linked List
2.Insert at Beginning
3.Insert After a key
4.Delete Before a Key
5.Display the Linked List
6.Exit
Enter a choice: 4
Enter key: 25
Deleted value-75
1.Create a Linked List
2.Insert at Beginning
3.Insert After a key
4.Delete Before a Key
5.Display the Linked List
6.Exit
Enter a choice:
```