```
//SINGLY LINKEDLIST
//EC23B1102
//MADHAMSHETTY SATHVIKA
#include <stdio.h>
#include <stdlib.h>
struct node{
   int data;
   struct node *next;
};
struct node *insert_node(struct node *head, int data, int c){
   int i=1;
   struct node *curr=head;
   struct node *temp = (struct node *)malloc(sizeof(struct
node));
   temp->data=data;
   temp->next=NULL;
   if(head==NULL)
      head=temp;
   else{
      if(c==1){
         temp->next=head;
         head=temp;
```

```
        }
        else if(c==2){
            int pos;
            printf("Enter the position to be inserted : \n");
            scanf("%d",&pos);
            while(i!=pos-1){
                curr=curr->next;
                i++;
                if(curr->next==NULL){
                    printf("sorry there are no such many nodes 
present\n");
                }}
            temp->next=curr->next;
            curr->next=temp;
        }
        else if(c==3){
            while(curr->next!=NULL){
                curr=curr->next;
            }
            curr->next=temp;
        }}
    return head;
```

```
}
struct node *delete_node(struct node *head, int c){
    int i=1;
    struct node *temp=head;
    struct node *prev=head;
    if(head==NULL)
        printf("Linkedlist is empty, nothing is there to delete\n");
    else{
        if(c==1){
            head=head->next;
            free(temp);
        }
        else if(c==2){
            int pos;
            printf("Enter the position to be deleted : \n");
            scanf("%d",&pos);
            for(i=0; i<pos; i++){
                if(i==0 && pos==1){
                    head=head->next;
                    free(temp);
                }
                else{
```

```
            if(i==pos-1 && temp!=NULL){
                prev->next=temp->next;
                free(temp);
            }
            else{
                prev=temp;
                if(prev==NULL){
                    break;}
                    temp=temp->next;
            }}}
        }
        else if(c==3){
            while(temp->next!=NULL){
                prev=temp;
                temp=temp->next;
            }
            free(temp);
        }}
    return head;
}
struct node *update(struct node *head, int data, int pos){
    int i=0;
    struct node *curr=head;
    while(curr!=NULL){
        if(i==pos-1){
            curr->data=data;
            break;
        }
        curr=curr->next;
        i++;
    }
    if(head==NULL)
        printf("Sorrry!\n");
    return head;
}
void search(struct node *head, int data){
    int i=0;
    while(head!=NULL){
        i++;
        if(head->data==data){
            printf("It is there in position : %d\n", i);
            break;
        }
        head=head->next;
    }
    if(head==NULL)
            printf("sorry!, it is not there\n");
}
void count(struct node *head){
    int i=0;
    while(head!=NULL){
        i++;
        head=head->next;
    }
    printf("count = %d\n", i);
}
void Printlist(struct node *head){
    while(head!=NULL){
        printf("%d\t", head->data);
        head=head->next;
    }}
void main(){
    struct node *head = NULL;
    int n, i, data, a, b, pos, c;
    printf("Enter number of nodes : \n");
    scanf("%d", &n);
    printf("Enter data : \n");
    for(i=0; i<n; i++){
        scanf("%d", &data);
        head = insert_node(head, data, 3);
    }
    Printlist(head);

printf("\na=1//insert.\na=2//delete.\na=3//search.\na=4//up
date.\na=5//count.");
    printf("Enter a: \n");
    scanf("%d", &a);
    switch (a){
        case 1:
            printf("Enter data to be inserted : \n");
            scanf("%d", &b);
            printf("c=1//Insert at beginning.\nc=2//Insert at a
position.\nc=3//Insert at end.\n");
            printf("Enter c : \n");
            scanf("%d", &c);
            head=insert_node(head, b, c);
            // printf("hi");
            break;
        case 2:
            printf("c=1//Delete at beginning.\nc=2//Delete at a
position.\nc=3//Delete at end.\n");
            printf("Enter c : \n");;
            scanf("%d", &c);
            head=delete_node(head, c);
            break;
        case 3:
            printf("Enter data to be searched : \n");
            scanf("%d", &b);
            search(head, b);
            break;
        case 4:
            printf("Enter data to updated : \n");
            scanf("%d", &b);
            printf("Enter the position to be updated : \n");
            scanf("%d", &pos);
            head=update(head, b, pos);
            break;
        case 5:
            count(head);
            break;
    }
```

```
    Printlist(head);
}
```

//DOUBLY LINKED LIST

//EC23B1102

//MADHAMSHETTY SATHVIKA

```
#include <stdio.h>
#include <stdlib.h>
struct node{
    int data;
    struct node *next, *prev;
};
struct node *insert_node(struct node *head, int data, int c){
    int i=1;
    struct node *curr=head;
    struct node *temp = (struct node *)malloc(sizeof(struct
node));
    temp->data=data;
    temp->next=NULL;
    temp->prev=NULL;
    if(head==NULL)
        head=temp;
    else{
```

```
if(c==1){
    temp->next=head;
    head->prev=temp;
    head=temp;
}
else if(c==2){
    int pos;
    printf("Enter the position to be inserted : \n");
    scanf("%d",&pos);
    while(i!=pos-1){
        curr=curr->next;
        i++;
        if(curr->next==NULL)
            printf("sorry");
    }
    temp->prev=curr;
    temp->next=curr->next;
    curr->next=temp;
    temp->next->prev=temp;
}
else if(c==3){
    while(curr->next!=NULL){
            curr=curr->next;}
          curr->next=temp;
          temp->prev=curr;
        }}
    return head;
}
struct node *delete_node(struct node *head, int c){
    int i=1;
    struct node *temp=head;
    if(head==NULL)
        head=temp;
    else if(head->next==NULL){
        temp=head;
        head=NULL;
        free(temp);
    }
    else{
        if(c==1){
            head->next->prev=NULL;
            temp=head;
            head=head->next;
            free(temp);
        }
        else if(c==2){
            int pos;
            printf("Enter the position of the data to be deleted :
\n");
            scanf("%d",&pos);
            while(i!=pos){
                temp=temp->next;
                i++;
                if(temp->next==NULL)
                    printf("sorry");
            }
            temp->prev->next=temp->next;
            temp->next->prev=temp->prev;
            free(temp);
        }
        else if(c==3){
            while(temp->next!=NULL){
                temp=temp->next;}
            temp->prev->next=NULL;
            free(temp);
        }}
```

```
    return head;
}
struct node *update(struct node *head, int data, int pos){
    int i=0;
    struct node *curr=head;
    while(curr!=NULL){
        if(i==pos-1){
            curr->data=data;
            break;
        }
        curr=curr->next;
        i++;
    }
    if(head==NULL)
        printf("Sorrry!\n");
    return head;
}
void search(struct node *head, int data){
    int i=0;
    while(head!=NULL){
        i++;
        if(head->data==data){
            printf("It is there in position : %d\n", i);
            break;
        }
        head=head->next;
    }
    if(head==NULL)
        printf("sorry!, it is not there\n");
}
void count(struct node *head){
    int i=0;
    while(head!=NULL){
        i++;
        head=head->next;
    }
    printf("count = %d\n", i);
}
void Printlist(struct node *head){
    while(head!=NULL){
        printf("%d\t", head->data);
        head=head->next;
    }}
void main(){
    struct node *head = NULL;
    int n, i, data, a, b, pos, c;
    printf("Enter number of nodes : \n");
    scanf("%d", &n);
    printf("Enter data : \n");
    for(i=0; i<n; i++){
        scanf("%d", &data);
        // printf("hi");
        head = insert_node(head, data, 3);
    }
    Printlist(head);

printf("\na=1//insert.\na=2//delete.\na=3//search.\na=4//up
date.\na=5//count.");
    printf("Enter a: \n");
    scanf("%d", &a);
    switch (a){
        case 1:
            printf("Enter data to be inserted : \n");
            scanf("%d", &b);
            printf("c=1//Insert at beginning.\nc=2//Insert at a
position.\nc=3//Insert at end.\n");
            printf("Enter c : \n");
            scanf("%d", &c);

            head=insert_node(head, b, c);

            break;

        case 2:

            printf("c=1//Delete at beginning.\nc=2//Delete at a
    position.\nc=3//Delete at end.\n");

            printf("Enter c : \n");;

            scanf("%d", &c);

            head=delete_node(head, c);

            break;

        case 3:

            printf("Enter data to be searched : \n");

            scanf("%d", &b);

            search(head, b);

            break;

        case 4:

            printf("Enter data to updated : \n");

            scanf("%d", &b);

            printf("Enter the position to be updated : \n");

            scanf("%d", &pos);

            head=update(head, b, pos);

            break;
        case 5:
            count(head);
            break;
    }
    Printlist(head);
}
```