Date:29.9.22

Program: To perform operations on Singly Linked list

Code:

```
#include<stdio.h>
#include<stdlib.h>
struct node
{	int data;
	struct node *next;
};
struct node *head=NULL, *ptr;
void insert_begin(int value)
{	struct node *temp;
	temp=(struct node *)malloc(sizeof(struct node));
	temp->data=value;
	temp->next=NULL;
	if (head== NULL)
	{			head=temp;
	}
	else
	{	temp->next=head;
		head=temp;
	}
}
void insert_end(int value)
{
	struct node *temp;
	temp=(struct node *)malloc(sizeof(struct node));
	temp->data=value;
	if (head== NULL)
	{
```

```
                head=temp;
        }
        else
        {
                ptr=head;
                while(ptr->next != NULL)
                {
                        ptr=ptr->next;
                }
                ptr->next=temp;
                temp->next=NULL;
        }
}
void insert_After_specified_ele(int value,int ele)
{
        struct node *temp;
        temp=(struct node *)malloc(sizeof(struct node));
        temp->data=value;
        if (head== NULL)
        {
                head=temp;
        }
        else
        {
                ptr=head;
                while(ptr->data != ele)
                {
                        ptr=ptr->next;
                }
                temp->next=ptr->next;
                ptr->next=temp;
```

```
        }}
void delete_begin()
{if (head== NULL)
        {
                printf("List if Empty, Deletion is not Possible");
        }
        else
        {       ptr=head;
                head=head->next;
                ptr->next=NULL;
                free(ptr);
        }
}
void delete_end()
{
        if (head== NULL)
        {
                printf("List if Empty, Deletion is not Possible");
        }
        else
        {
                struct node *p;
                ptr=head;
                while(ptr->next != NULL)
                {
                p=ptr;
                ptr=ptr->next;
                }
                p->next=NULL;
                free(ptr);
        }
```

```
}
void delete_middle(int ele)
{
	if (head== NULL)
	{
		printf("List if Empty, Deletion is not Possible");
	}
	else
	{
		struct node *p;
		ptr=head;
		while(ptr->data != ele)
		{
			p=ptr;
			ptr=ptr->next;
		}
		p->next=ptr->next;
		free(ptr);
	}
}
void display()
{
	if (head== NULL)
	{
		printf("List if Empty");
	}
	else
	{
		ptr=head;
		while( ptr != NULL)
		{
```

```
                printf("-> %d", ptr->data);
                ptr=ptr->next;
            }
            printf("\n");
        }
}
void main()
{
    int value,element,choice;
    printf("1. Insert_begin \n");
    printf("2. Insert_end \n");
    printf("3. Insert_After_specified_element \n");
    printf("4. delete_begin \n");
    printf("5. delete_end \n");
    printf("6. delete_middle \n");
    printf("7. exit \n");
    while(1)
    {
        printf("Enter the Choice: ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:
                {
                    printf("Enter the value");
                    scanf("%d",&value);
                    insert_begin(value);
                    display();
                    break;
                }
                case 2:
```

```
{
        printf("Enter the value");
        scanf("%d",&value);
        insert_end(value);
        display();
        break;
}
case 3:
        {
        printf("Enter the value");
        scanf("%d",&value);
        printf("After which element u want to insert");
        scanf("%d",&element);
        insert_After_specified_ele(value,element);
        display();
        break;
}
case 4:
        {
        delete_begin();
        display();
        break;
}
case 5:
{
        delete_end();
        display();
        break;
}
case 6:
{
```

```
                printf("Enter the element you want to delete");
                scanf("%d",&element);
                delete_middle(element);
                display();
                break;
            }
            case 7:
                exit(0);

            }
        }
    }
```

Output:

```
1. Insert_begin
2. Insert_end
3. Insert_After_specified_element
4. delete_begin
5. delete_end
6. delete_middle
7. exit
Enter the Choice: 1
Enter the value4
-> 4
Enter the Choice: 1
Enter the value8
-> 8-> 4
Enter the Choice: 2
Enter the value6
-> 8-> 4-> 6
Enter the Choice: 6
Enter the element you want to delete4
-> 8-> 6
Enter the Choice: 7
```