

DS PRACTICAL No. 04

AIM:- Implement a singly linked list and perform the operation like insertion, deletion and traversal.

PROGRAM:-

1] Insert a node at the beginning (LinkedList)

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

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

```
struct node
```

```
{
```

```
    int data;
```

```
    struct node *next;
```

```
};
```

```
int main()
```

```
{
```

```
    struct node * s,*p,*q;
```

```
    int ch;
```

```
    p=(struct node *)malloc(sizeof(struct node));
```

```
    printf("\nEnter data of first node :");
```

```
    scanf("%d",&p->data);
```

```
    s=p;
```

```
    do{
```

```
        q=(struct node *)malloc(sizeof(struct node));
```

```
        printf("\nEnter data of next node :");
```

```
        scanf("%d",&q->data);
```

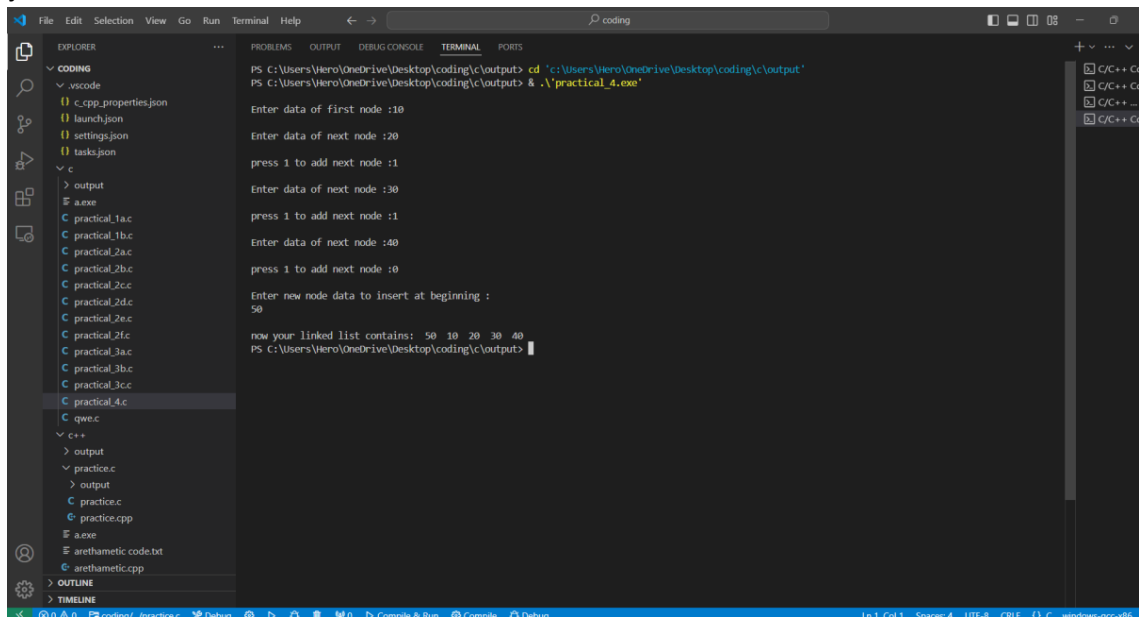
```
        p->next=q;
```

```
p=q;
printf("\npress 1 to add next node :");
scanf("%d",&ch);
; } while(ch==1);
```

```
p->next=NULL;
struct node * a;
a=(struct node *)malloc(sizeof(struct node));
printf("\nEnter new node data to insert at beginning :\n");
scanf("%d",&a->data);
a->next=s;
s=a;
printf("\nnow your linked list contains: ");
while(s!=NULL)
{

printf(" %d ",s->data);
s=s->next;
}
```

}



```
PS C:\Users\Hero\OneDrive\Desktop\coding\c\output> cd 'c:\Users\Hero\OneDrive\Desktop\coding\c\output'
PS C:\Users\Hero\OneDrive\Desktop\coding\c\output> & .\practical_4.exe

Enter data of first node :10

Enter data of next node :20

press 1 to add next node :1

Enter data of next node :30

press 1 to add next node :1

Enter data of next node :40

press 1 to add next node :0

Enter new node data to insert at beginning :
50

now your linked list contains: 50 10 20 30 40
PS C:\Users\Hero\OneDrive\Desktop\coding\c\output>
```

2. Insert a node at the end (LinkedList)

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

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

```
struct node{
```

```
    int data;
```

```
    struct node * next;
```

```
};
```

```
int main()
```

```
{ struct node * p,*q,*s,*first;
```

```
    int ch;
```

```
    p=(struct node *)malloc(sizeof(struct node));
```

```
    printf("Enter first node data :");
```

```
    scanf("%d",&p->data);
```

```
    s=p;
```

```

// s points to the first node
first=s;

// first keeps track of the start of the list
do{ q=(struct node *)malloc(sizeof(struct node));
printf("Enter next node data :");
scanf("%d",&q->data);
p->next=q;
p=q;
printf("\npress 1 for next node :");
scanf("%d",&ch);
}while(ch==1);
p->next=NULL;

// Insert a node at the end
struct node * a;
a=(struct node *)malloc(sizeof(struct node));
printf("\nEnter node data to insert at end :");
scanf("%d",&a->data);
while(s->next!=NULL){
s=s->next;
}
s->next=a;
a->next=NULL;
s=first;
printf("\nnow your linked list contains: ");
while(s!=NULL)
{

```

```

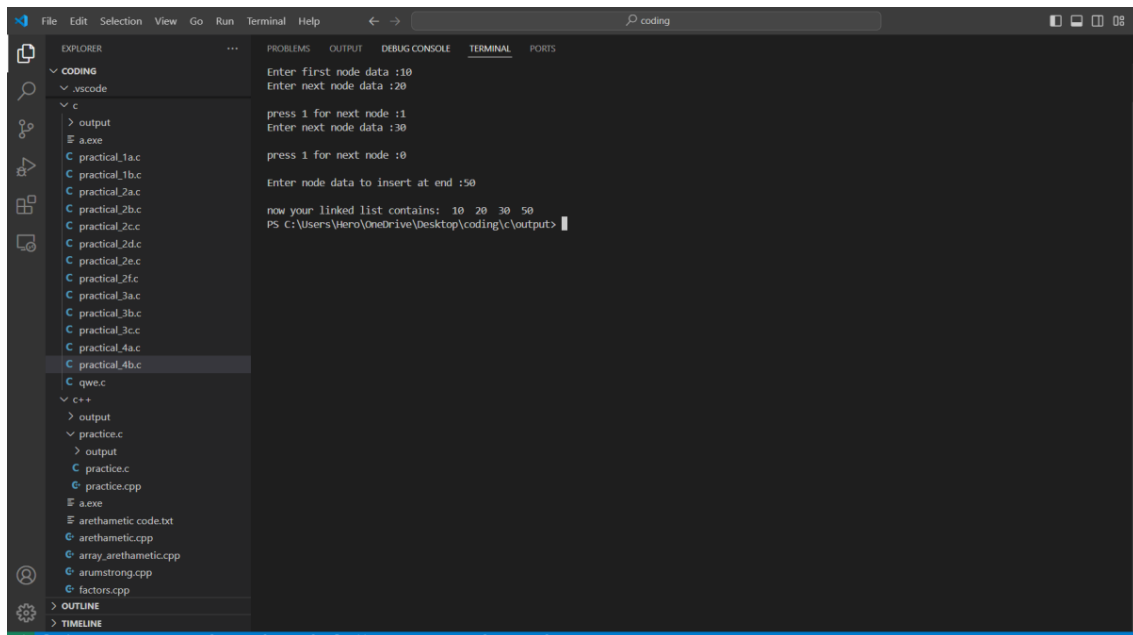
printf(" %d ",s->data);

s=s->next;

}

}

```



3. Delete a node at the beginning (LinkedList)

```

#include<stdio.h>

#include<stdlib.h>

struct node{

    int data;

    struct node * next;

};

int main()

```

```

{
    struct node *s,*p,*q,*first;
    char ch;
    p=(struct node *)malloc(sizeof(struct node));
    printf("\nEnter first node data : ");
    scanf("%d",&p->data);
    s=p;
    // s points to the first node
    first =s ;
    // first keeps track of the start of the list
    do{
        q=(struct node *)malloc(sizeof(struct node));
        printf("Enter next node data : ");
        scanf("%d",&q->data);
        p->next=q;
        p=q;
        printf("\npress 1 for next node : ");
        scanf("%d",&ch); }while(ch==1);
        p->next=NULL;
        if(s==NULL)
        {
            printf("Underflow");
        }
        s=first;
        first=first->next;
        free(s);
    }
}

```

```
s=first;
```

```
printf("\n\nafter delete 1st node linked list contains: ");
```

```
while(s!=NULL)
```

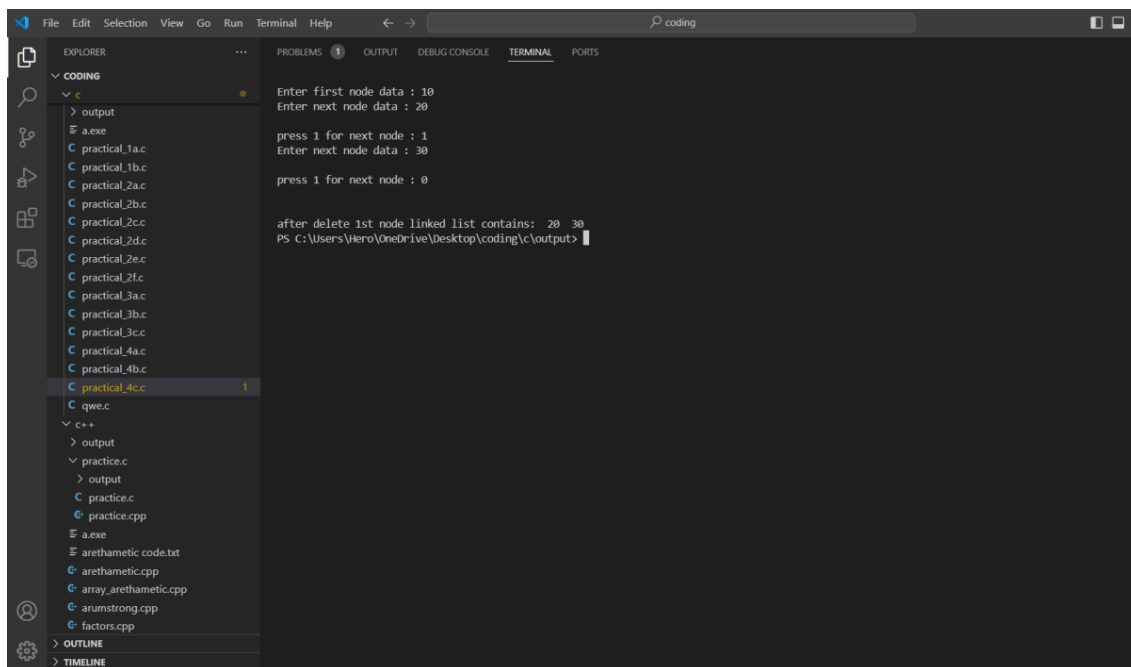
```
{
```

```
printf(" %d ",s->data);
```

```
s=s->next;
```

```
}
```

```
}
```



```
File Edit Selection View Go Run Terminal Help
coding

EXPLORER
CODING
  c
    output
    a.exe
    practical_1a.c
    practical_1b.c
    practical_2a.c
    practical_2b.c
    practical_2c.c
    practical_2d.c
    practical_2e.c
    practical_2f.c
    practical_3a.c
    practical_3b.c
    practical_3c.c
    practical_4a.c
    practical_4b.c
    practical_4c.c
    qwe.c
  c++
    output
    practice.c
    practice.c
    practice.cpp
    a.exe
    arethametic code.txt
    arethametic.cpp
    array_arethametic.cpp
    arumstrong.cpp
    factors.cpp
  OUTLINE
  TIMELINE

TERMINAL
Enter first node data : 10
Enter next node data : 20

press 1 for next node : 1
Enter next node data : 30

press 1 for next node : 0

after delete 1st node linked list contains: 20 30
PS C:\Users\Hero\OneDrive\Desktop\coding\c\output>
```

4. Delete a node at the end (LinkedList)

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

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

```
struct node{
```

```
int data;
```

```

    struct node * next;
};

int main()
{
    struct node *s,*p,*q,*first;
    int ch;
    p=(struct node *)malloc(sizeof(struct node));
    printf("\nEnter first node data : ");
    scanf("%d",&p->data);
    s=p;
    // s points to the first node
    first =s ;
    // first keeps track of the start of the list
    do{
        q=(struct node *)malloc(sizeof(struct node));
        printf("Enter next node data : ");
        scanf("%d",&q->data);
        p->next=q;
        p=q;
        printf("\npress 1 for next node : ");
        scanf("%d",&ch); }while(ch==1);
    p->next=NULL;
    while(s->next!=NULL)
    {
        p=s;
        s=s->next;
    }
}

```



```

}

p->next=NULL;

free(s);

s=first;

printf("\nafter delete last node linked list contains: ");

while(s!=NULL)

{

printf(" %d ",s->data);

s=s->next;

}

}

```

```

Enter next node data : 20
press 1 for next node : 1
Enter next node data : 30
press 1 for next node : 1
Enter next node data : 50
press 1 for next node : 0
after delete last node linked list contains: 10 20 30
PS C:\Users\Viern\OneDrive\Desktop\coding\c\output>

```

5. Delete a node at the any Position (LinkedList)

```
#include<stdio.h>

#include<stdlib.h>

struct node{

    int data;

    struct node * next;

};

int main()

{

    struct node *s,*p,*q,*first;

    int ch;

    int key;

    p=(struct node *)malloc(sizeof(struct node));

    printf("\nEnter first node data : ");

    scanf("%d",&p->data);

    s=p;

    // s points to the first node

    first =s ;

    // first keeps track of the start of the list

    do{

        q=(struct node *)malloc(sizeof(struct node));

        printf("Enter next node data : ");

        scanf("%d",&q->data);

        p->next=q;

        p=q;

        printf("\npress 1 for next node : ");

        scanf("%d",&ch); }while(ch==1);
```

```
p->next=NULL;
printf("Enter data of node to delete : ");
scanf("%d",&key);
while(s->data!=key)
{
p=s;
s=s->next;
}
p->next=s->next;
free(s);
s=first;
printf("\nafter deleting node linked list contains: ");
while(s!=NULL)
{

printf(" %d ",s->data);
s=s->next;
}
}
```

