

## DS PRACTICAL NO : 4

AIM : implement a single linked list and perform the operation like insertion, deletion and traversal.

PROGRAM :

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

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

```
#include <stdbool.h>
```

```
struct node
```

```
{
```

```
    int data;
```

```
    struct node *next;
```

```
    struct node *priv;
```

```
};
```

```
int main()
```

```
{
```

```
    struct node *a, *b, *c, *d, *e, *f, *g;
```

```
    int search;
```

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

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

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

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

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

## DS PRACTICAL NO : 4

```
f = (struct node *)malloc(sizeof(struct node))
```

```
a->data = 10;
```

```
b->data = 20;
```

```
c->data = 30;
```

```
d->data = 40;
```

```
e->data = 50;
```

```
a->next =
```

```
b->next = c;
```

```
c->next = d;
```

```
d->next = e;
```

```
e->next = NULL;
```

```
struct node *p = a;
```

```
printf("Traversing the linked list in forward direction: \n");
```

```
while (p != NULL)
```

```
{
```

```
    printf("%d\t", p->data);
```

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

```
}
```

```
struct node *m = e;
```

```
printf("\n");
```

```
m = a;
```

## DS PRACTICAL NO : 4

```
printf("Enter the node after which you have to insert the data:\n");
```

```
scanf("%d", &search);
```

```
while (m != NULL && m->data != search)
```

```
{
```

```
    m = m->next;
```

```
};
```

```
if (m->data == search)
```

```
{
```

```
    printf("Enter the data of the new node:\n");
```

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

```
    f->next = m->next;
```

```
    f -> prev = m;
```

```
    m->next = f;
```

```
}else
```

```
{
```

```
    printf("The searching data not found\n");
```

```
}
```

```
m = a;
```

```
while (m != NULL)
```

```
{
```

```
    printf("%d\t", m->data);
```

```
    m = m->next;
```

```
}
```

## DS PRACTICAL NO : 4

```
printf("\n");
m = a;
printf("Enter the node which you have to DELETE:\n");
scanf("%d", &search);
while (m != NULL && m->data != search)
{
    g = m;
    m = m->next;
}

if (m->data == search)
{
    g->next = m->next;
    free(m);
}
else
{
    printf("The data not found.\n");
}

m = a;
while (m != NULL)
{
    printf("%d\t", m->data);
    m = m->next;
}

return 0;
}\
```

# DS PRACTICAL NO : 4

## OUTPUT

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS C:\Users\230212\Desktop\DSA_programs> cd "c:\Users\230212\Desktop\DSA_programs"
PS C:\Users\230212\Desktop\DSA_programs> cd "c:\Users\230212\Desktop\DSA_programs\" ; if ($?) { gcc Practical4.c -o Practical4 } ; if ($?) { .\Practical4
Traversing the linked list in forward direction:
10    20    30    40    50
Enter the node after which you have to insert the data:
20
Enter the data of the new node:
45
10    20    45    30    40    50
Enter the node which you have to DELETE:
30
10    20    45    40    50
PS C:\Users\230212\Desktop\DSA_programs> |
```

GITHUB LINK : <https://github.com/AmolNagargoje04/Data-Structure-practical>