DSA Practical No -4

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

Program:

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

#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));

f = (struct node \*)malloc(sizeof(struct node));

   a->data = 21;    b->data = 22;

    c->data = 23;     d->data = 8;     e->data = 300;

    a->next = b;

    b->next = c;

    c->next = d;

    d->next = e;

    e->next = NULL;

struct node \*p = a;

printf("Traversal of 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;

    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 -> priv = 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;

    }

    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)

    {

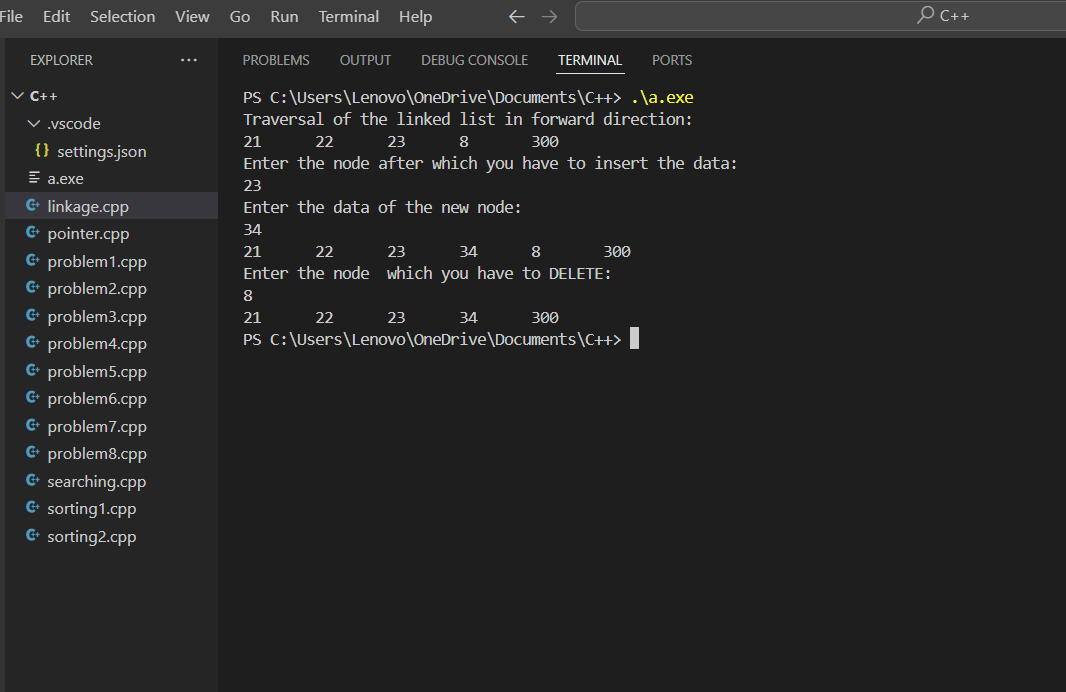
        printf("%d\t", m->data);         m = m->next;

    }

    return 0;

}

Output:



Github link:

<https://github.com/Manas1597/DSA>