**DOUBLY LINKED LIST**

**NAME: ANISH NAPHADE DIV: AI A ROLL NO: 07**

**PRN: 12211542**

#include <stdio.h>

#include <stdlib.h>

void createNode();

void displayfromend();

void deletebeg();

void deleteend();

void displayfromstart();

struct node

{

    int data;

    struct node \*next;

    struct node \*prev;

};

struct node \*head, \*temp;

void main()

{

int choose;

int choose1=1;

while(choose1){

    printf("press given number to perform operation :\n 1.create node\n 2.display data from start\n 3. display the data reverse\n 4.delete from beginning\n 5.delete from end \n" );

    scanf("%d", &choose);

    switch (choose)

    {

    case 1:

    createNode();

        break;

        case 2:displayfromstart();

        break;

        case 3:displayfromend();

        break;

        case 4:deletebeg();

        break;

        case 5:deleteend();

        break;

    default:

    printf("enter a valid number");

        break;

    }

    printf("\nto continue press 1 =\n");

    scanf("%d",&choose1);}

}

void createNode()

{

    head = 0;

    temp = head;

    struct node \*newnode;

    int choice = 1;

    while (choice)

    {

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

        printf("enter the data=");

        scanf("%d", &newnode->data);

        newnode->next = 0;

        newnode->prev = 0;

        if (head == 0)

        {

            head = temp = newnode;

        }

        else

        {

            temp->next = newnode;

            newnode->prev = temp;

            temp = newnode;

        }

        printf("if you want to continue press 1=");

        scanf("%d", &choice);

    }

}

void displayfromend()

{

    if (head == 0)

    {

        printf("list is empty\n");

    }

    else

    {

        temp = head;

    while (temp->next != 0)

    {

        temp = temp->next;

    }

        while (temp != 0)

        {

            printf("\n %d", temp->data);

            temp = temp->prev;

        }

    }

}

void displayfromstart()

{temp=head;

    if (head == 0)

    {

        printf("list is empty\n");

    }

    else

    {

        while (temp != 0)

        {

            printf("\n %d", temp->data);

            temp = temp->next;

        }

    }

}

void deletebeg()

{

    struct node \*tempfree;

    if (head == 0)

    {

        printf("list is empty\n");

    }

    else

    {

        head = head->next;

        head->prev = 0;

    }

    free(tempfree);

}

void deleteend()

{

    struct node \*temp1;

    temp = head;

    while (temp->next != 0)

    {

        temp = temp->next;

    }

    temp1 = temp;

    temp = temp->prev;

    temp->next = 0;

    free(temp1);

}