**TITLE 44**

Write a C program to create a doubly linked list with 5 nodes

**OBJECTIVE:**

By the end of this problem we will be able to create a doubly linked list with 5 nodes

**PROBLEM STATEMENT:**

In this program we create a doubly linked list. Input from user:

Input the number of nodes :

Once the input is collected and stored the output is printed.

**ALGORITHM:**

START

Define variables: num, \* preptr, \* nextptr, \*strnode, \*ennode

INPUT: Read from the user

COMPUTATION: Computing the doubly linked list

DISPLAY: Displaying the doubly linked list

STOP

**PROGRAM:**

#include <stdio.h>

#include <stdlib.h>

struct node {

int num;

struct node \* preptr;

struct node \* nextptr;

}\*stnode, \*ennode;

void DlListcreation(int n);

void displayDlList();

int main()

{

int n;

stnode = NULL;

ennode = NULL;

printf("\n\n Doubly Linked List : Create and display a doubly linked list :\n");

printf("-------------------------------------------------------------------\n");

printf(" Input the number of nodes : ");

scanf("%d", &n);

DlListcreation(n);

displayDlList();

return 0;

}

void DlListcreation(int n)

{

int i, num;

struct node \*fnNode;

if(n >= 1)

{

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

if(stnode != NULL)

{

printf(" Input data for node 1 : "); // assigning data in the first node

scanf("%d", &num);

stnode->num = num;

stnode->preptr = NULL;

stnode->nextptr = NULL;

ennode = stnode;

// putting data for rest of the nodes

for(i=2; i<=n; i++)

{

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

if(fnNode != NULL)

{

printf(" Input data for node %d : ", i);

scanf("%d", &num);

fnNode->num = num;

fnNode->preptr = ennode; // new node is linking with the previous node

fnNode->nextptr = NULL;

ennode->nextptr = fnNode; // previous node is linking with the new node

ennode = fnNode; // assign new node as last node

}

else

{

printf(" Memory can not be allocated.");

break;

}

}

}

else

{

printf(" Memory can not be allocated.");

}

}

}

void displayDlList()

{

struct node \* tmp;

int n = 1;

if(stnode == NULL)

{

printf(" No data found in the List yet.");

}

else

{

tmp = stnode;

printf("\n\n Data entered on the list are :\n");

while(tmp != NULL)

{

printf(" node %d : %d\n", n, tmp->num);

n++;

tmp = tmp->nextptr; // current pointer moves to the next node

}

}

}

**CONCLUSION:**

The simulation of the above C program helps us understand how a doubly linked list can be created.

**OUTPUT:**

Doubly Linked List : Create and display a doubly linked list :

-------------------------------------------------------------

Input the number of nodes : 5

Input data from node 1 : 5

Input data from node 2 : 6

Input data from node 3 : 7

Input data from node 4 : 8

Input data from node 5 : 9

Data entered in the list are :

Data 1 = 5

Data 2 = 6

Data 3 = 7

Data 4 = 8

Data 5 = 9