**TITLE 41**

Write a C program to create a single linked list with 5 nodes. (5 integers are taken from user input) and display the linked-list elements

**OBJECTIVE:**

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

**PROBLEM STATEMENT:**

In this program we create a singly 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, \*nextptr, \*stnode

INPUT: Read from the user

COMPUTATION: Computing the singly linked list

DISPLAY: Displaying the singly linked list

STOP

**PROGRAM:**

#include <stdio.h>

#include <stdlib.h>

struct node

{

int num; //Data of the node

struct node \*nextptr; //Address of the next node

}\*stnode;

void createNodeList(int n); // function to create the list

void displayList(); // function to display the list

int main()

{

int n;

printf("\n\n Linked List : To create and display Singly Linked List :\n");

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

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

scanf("%d", &n);

createNodeList(n);

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

displayList();

return 0;

}

void createNodeList(int n)

{

struct node \*fnNode, \*tmp;

int num, i;

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

if(stnode == NULL) //check whether the fnnode is NULL and if so no memory allocation

{

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

}

else

{

// reads data for the node through keyboard

printf(" Input data for node 1 : ");

scanf("%d", &num);

stnode->num = num;

stnode->nextptr = NULL; // links the address field to NULL

tmp = stnode;

// Creating n nodes and adding to linked list

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

{

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

if(fnNode == NULL)

{

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

break;

}

else

{

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

scanf(" %d", &num);

fnNode->num = num; // links the num field of fnNode with num

fnNode->nextptr = NULL; // links the address field of fnNode with NULL

tmp->nextptr = fnNode; // links previous node i.e. tmp to the fnNode

tmp = tmp->nextptr;

}

}

}

}

void displayList()

{

struct node \*tmp;

if(stnode == NULL)

{

printf(" List is empty.");

}

else

{

tmp = stnode;

while(tmp != NULL)

{

printf(" Data = %d\n", tmp->num); // prints the data of current node

tmp = tmp->nextptr; // advances the position of current node

}

}

}

**CONCLUSION:**

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

**OUTPUT:**

Linked List : To create and display Singly Linked List :

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

Input the number of nodes : 5

Input data from node 1 : 10

Input data from node 2 : 15

Input data from node 3 : 20

Input data from node 4 : 25

Input data from node 5 : 30

Data entered in the list are :

Data = 10

Data = 15

Data = 20

Data = 25

Data = 30