**Creation of Linked List**

#include <stdio.h>

#include <stdlib.h>

struct node

{

int num; //Data of the node

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

}\*head;

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

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

int main()

{

int n;

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

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

scanf("%d", &n);

create(n);

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

display();

return 0;

}

void create(int n)

{

struct node \*newnode, \*tmp;

int num, i;

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

if(head == NULL) //check whether the node 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);

head->num = num;

head->next = NULL; // links the address field to NULL

tmp = head;

// Creating n nodes and adding to linked list

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

{

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

if(newnode == NULL)

{

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

break;

}

else

{

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

scanf(" %d", &num);

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

newnode->next = NULL; // links the address field of newnode with NULL

tmp->next = newnode; // links previous node i.e. tmp to the newnode

tmp = tmp->next;

}

}

}

}

void displayList()

{

struct node \*tmp;

if(head == NULL)

{

printf(" List is empty.");

}

else

{

tmp = head;

while(tmp != NULL)

{

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

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

}

}

}