Name: MD RAKIBUL ISLAM

ID: 20183290424

Data structure Homework

Experiment Part

Session3-Part1

Create a linked list. Enter the data from the keyboard (Reverse order).

```c
#include <stdio.h>
#include <stdlib.h>

struct node
{
    int num;                    //Data of the node
    struct node *nextptr;       //Address of the node
}*stnode;

void createNodeList(int n);     //function to create the list
void reverseDispList();         //function to convert the list in reverse
void displayList();             //function to display the list

int main()
{
    int n;
        printf("\n\n Linked List : Create a singly linked list and print it in reverse order :\n");
    printf(" Input the number of nodes : ");
    scanf("%d", &n);
    createNodeList(n);
    printf("\n Data entered in the list are : \n");
    displayList();
    reverseDispList();
    printf("\n The list in reverse are :  \n");
    displayList();
    return 0;
}
```

```
29  void createNodeList(int n)
30  {
31      struct node *fnNode, *tmp;
32      int num, i;
33      stnode = (struct node *)malloc(sizeof(struct node));
34      if(stnode == NULL) //check whether the stnode is NULL and if so no memory allocation
35      {
36          printf(" Memory can not be allocated.");
37      }
38      else
39      {
40  // reads data for the node through keyboard
41          printf(" Input data for node 1 : ");
42          scanf("%d", &num);
43          stnode-> num = num;
44          stnode-> nextptr = NULL; //Links the address field to NULL
45          tmp = stnode;
46  //Creates n nodes and adds to linked list
47          for(i=2; i<=n; i++)
48          {
49              fnNode = (struct node *)malloc(sizeof(struct node));
50              if(fnNode == NULL) //check whether the fnnode is NULL and if so no memory allocation
51              {
52                  printf(" Memory can not be allocated.");
53                  break;
54              }
55              else
55              else
56              {
57                  printf(" Input data for node %d : ", i);
58                  scanf(" %d", &num);
59                  fnNode->num = num;      // links the num field of fnNode with num
60                  fnNode->nextptr = NULL; // links the address field of fnNode with NULL
61                  tmp->nextptr = fnNode; // links previous node i.e. tmp to the fnNode
62                  tmp = tmp->nextptr;
63              }
64          }
65      }
66  }
67
68  void reverseDispList()
69  {
70      struct node *prevNode, *curNode;
71
72      if(stnode != NULL)
73      {
74          prevNode = stnode;
75          curNode = stnode->nextptr;
76          stnode = stnode->nextptr;
77
78          prevNode->nextptr = NULL; //convert the first node as last
79
```

```
        while(stnode != NULL)
        {
            stnode = stnode->nextptr;
            curNode->nextptr = prevNode;

            prevNode = curNode;
            curNode = stnode;
        }
        stnode = prevNode; //convert the last node as head
    }
}

void displayList()
{
    struct node *tmp;
    if(stnode == NULL)
    {
        printf(" No data found in the list.");
    }
    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
        }
    }
}
```

Result:

Output will be