Name: MD RAKIBUL ISLAM

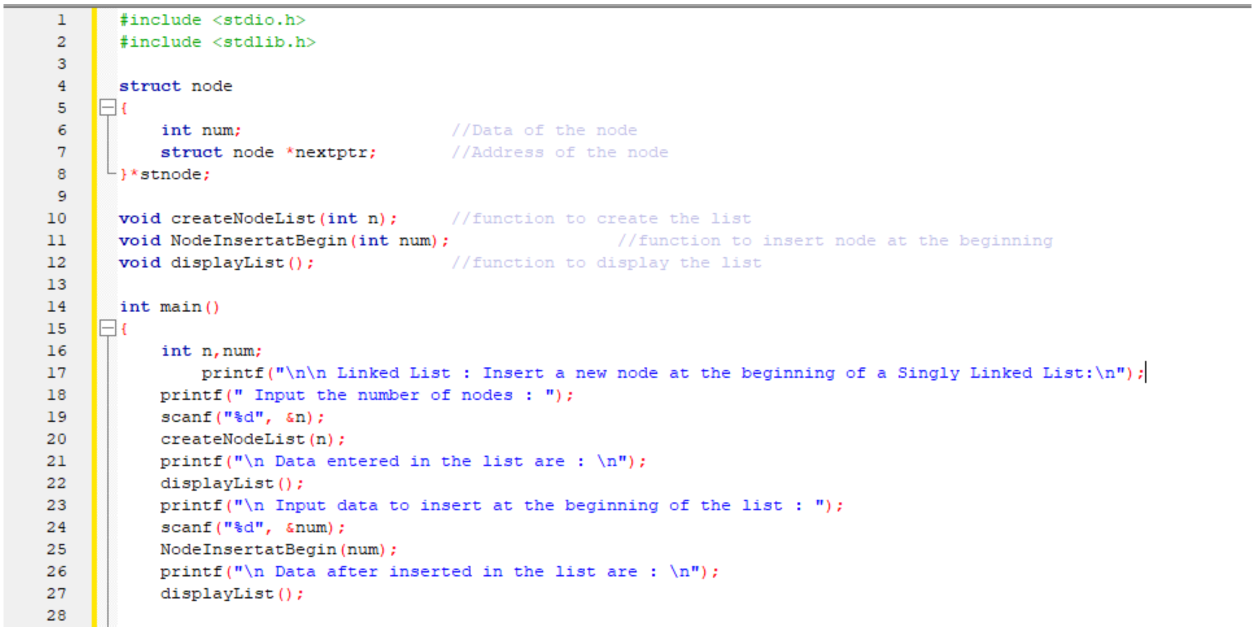
ID: 20183290424

Data structure Homework

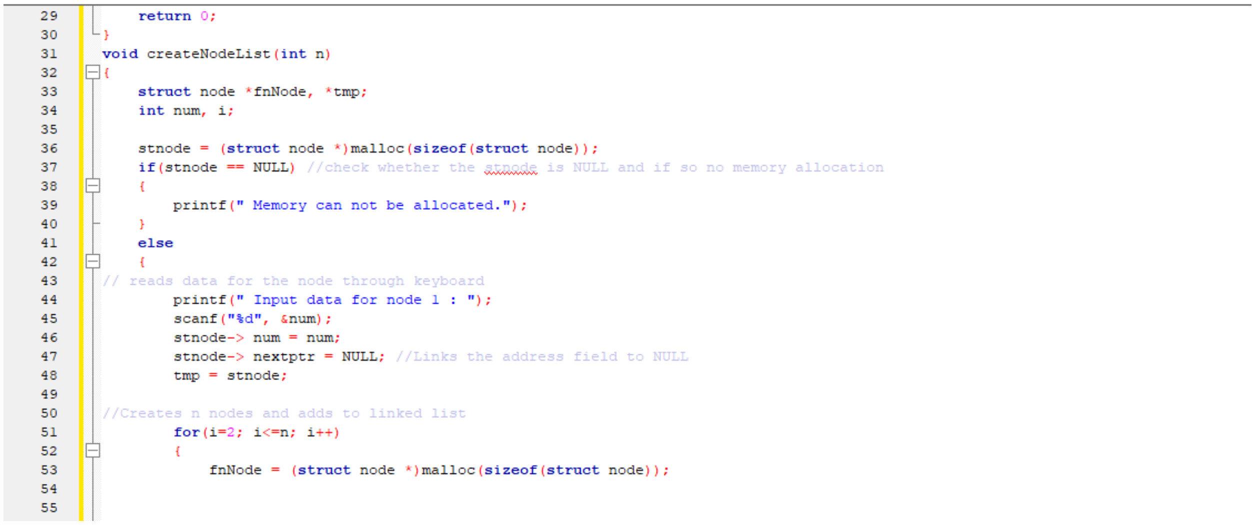
Experiment Part

Session3-Part1

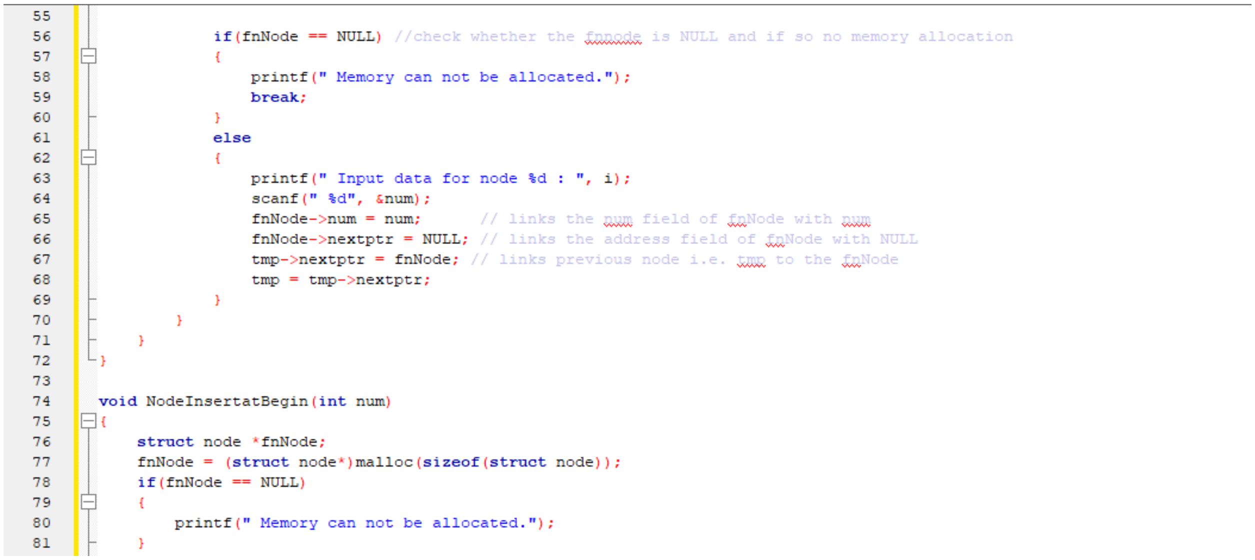
Create a linked list. Enter the data from the keyboard (Head insertion).



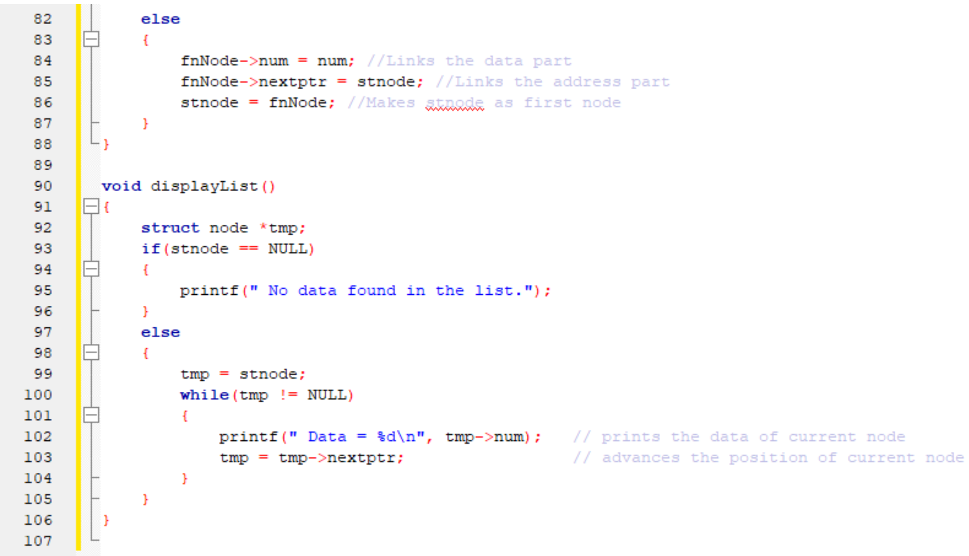
```
#include <stdio.h>
#include <stdlib.h>

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 NodeInsertatBegin(int num);                //function to insert node at the beginning
void displayList();             //function to display the list

int main()
{
    int n,num;
        printf("\n\n Linked List : Insert a new node at the beginning of a Singly Linked List:\n");
    printf(" Input the number of nodes : ");
    scanf("%d", &n);
    createNodeList(n);
    printf("\n Data entered in the list are : \n");
    displayList();
    printf("\n Input data to insert at the beginning of the list : ");
    scanf("%d", &num);
    NodeInsertatBegin(num);
    printf("\n Data after inserted in the list are : \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 stnode 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;

//Creates n nodes and adds to linked list
        for(i=2; i<=n; i++)
        {
            fnNode = (struct node *)malloc(sizeof(struct node));


```



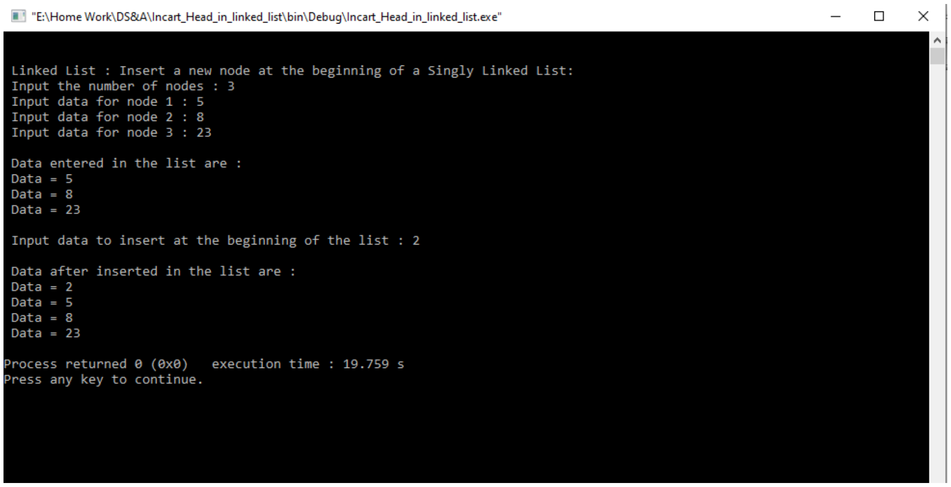
```
            if(fnNode == NULL) //check whether the fnnode is NULL and if so no memory allocation
            {
                printf(" Memory can not be allocated.");
                break;
            }
            else
            {
                printf(" Input data for node %d : ", i);
                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 NodeInsertatBegin(int num)
{
    struct node *fnNode;
    fnNode = (struct node*)malloc(sizeof(struct node));
    if(fnNode == NULL)
    {
        printf(" Memory can not be allocated.");
    }
```



```
    else
    {
        fnNode->num = num; //Links the data part
        fnNode->nextptr = stnode; //Links the address part
        stnode = fnNode; //Makes stnode as first node
    }
}

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



"E:\Home Work\DS&A\Incart_Head_in_linked_list\bin\Debug\Incart_Head_in_linked_list.exe"

```
 Linked List : Insert a new node at the beginning of a Singly Linked List:
 Input the number of nodes : 3
 Input data for node 1 : 5
 Input data for node 2 : 8
 Input data for node 3 : 23

 Data entered in the list are :
 Data = 5
 Data = 8
 Data = 23

 Input data to insert at the beginning of the list : 2

 Data after inserted in the list are :
 Data = 2
 Data = 5
 Data = 8
 Data = 23

Process returned 0 (0x0)   execution time : 19.759 s
Press any key to continue.
```