

<b>Subject:</b> - DSU	<b>Subject Code:</b> 313301
<b>Semester:</b> - III	<b>Course:</b> Computer Engineering
Laboratory No: L003	<b>Name of Subject Teacher:</b> Prof. Imran S.
<b>Name of Student:</b> Mohd Saad Khan	<b>Roll Id:</b> - 24203A0007
<b>Experiment No:</b>	12
<b>Title of Experiment</b>	* Write a 'C' Program to Implement Singly Linked List with Operations: (i) Insert at beginning, (ii) Search, (iii) Display

**Aim:** Write a 'C' Program to Implement Singly Linked List with Operations: (i) Insert at beginning, (ii) Search, (iii) Display

### Algorithm:

#### Algorithm for Linked List Program

Step 1: Start

Step 2: Define a structure Node with two members:

- data (integer)
- next (pointer to next node)

Step 3: Declare functions:

- createlinkedlist(n) – to create a linked list of n nodes
- createnode(data) – to create a new node
- printList(head) – to display the linked list
- begin(head) – to insert a node at the beginning
- search(head) – to search an element in the linked list

Step 4: In main():

- Read number of nodes n
- Call createlinkedlist(n) and store the returned head pointer
- Call printList(head) to display the list
- Call begin(head) to insert a new node at the beginning and update head
- Call printList(head) again to show updated list
- Call search(head) to find an element

#### Function: createlinkedlist(n)

Step 1: If  $n \leq 0$ , print error and return NULL

Step 2: Read data for first node

- Call createnode(data) to create the node
- Assign it to head and temp

Step 3: Repeat from i = 2 to n

- Read data for next node
- Create new node using createnode(data)
- Link temp->next = newNode
- Move temp = newNode

Step 4: Return head pointer

**Function: createnode(data)**

Step 1: Allocate memory dynamically for a new node

Step 2: Assign newNode->data = data

Step 3: Set newNode->next = NULL

Step 4: Return pointer to newNode

**Function: printList(head)**

Step 1: Set temp = head

Step 2: While temp != NULL

a) Print temp->data

b) Move to next node temp = temp->next

Step 3: Print "NULL" at the end

**Function: begin(head)**

Step 1: Read data for new node

Step 2: Create new node using createnode(data)

Step 3: Point newNode->next = head

Step 4: Return newNode as new head

**Function: search(head)**

Step 1: Read element data to search

Step 2: Initialize counter = 1, temp = head

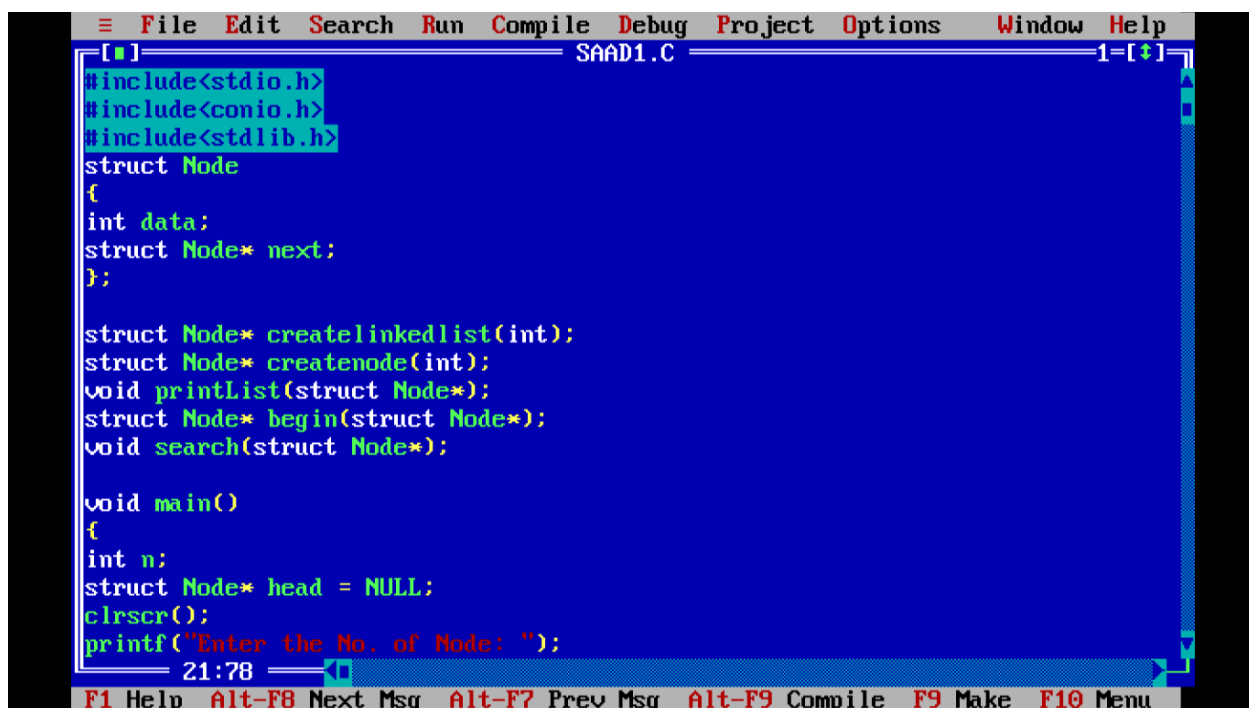
Step 3: Traverse while temp != NULL

a) If temp->data == data, print that data is found at position counter

b) Otherwise move to next node and increment counter

Step 4: If loop ends without match, print "Element not found"

**Code:**



```
SAAD1.C
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct Node
{
    int data;
    struct Node* next;
};

struct Node* createlinkedlist(int);
struct Node* createnode(int);
void printList(struct Node*);
struct Node* begin(struct Node*);
void search(struct Node*);

void main()
{
    int n;
    struct Node* head = NULL;
    clrscr();
    printf("Enter the No. of Node: ");
    21:78
```

```
≡ File Edit Search Run Compile Debug Project Options Window Help
[■] SAAD1.C 1-[↵]
scanf("%i",&n);
head = createlinkedlist(n);
printList(head);
head = begin(head);
printList(head);
search(head);
getch();
}

struct Node* createlinkedlist(int n)
{
int data,i;
struct Node *head = NULL, *temp = NULL, *newNode = NULL;
if(n<=0)
{
printf("Number of Nodes should be greater then Zero...");
return NULL;
}
printf("Enter data for Node 1: ");
scanf("%i",&data);
newNode = createnode(data);
42:78

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu
≡ File Edit Search Run Compile Debug Project Options Window Help
[■] SAAD1.C 1-[↵]
head = newNode;
temp = newNode;
for(i=2;i<=n;i++)
{
printf("Enter data for Node %i: ",i);
scanf("%i",&data);
newNode = createnode(data);
temp->next = newNode;
temp = newNode;
}
return head;
}

struct Node* createnode(int data)
{
struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
if(!newNode)
{
printf("Memory Allocation Error...");
}
newNode->data = data;
63:78

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu
```

```
≡ File Edit Search Run Compile Debug Project Options Window Help
[■] SAAD1.C 1-[↕]
newNode->next = NULL;
return newNode;
}

void printList(struct Node* head)
{
    struct Node* temp = head;
    while(temp!=NULL)
    {
        printf("%i -> ",temp->data);
        temp = temp->next;
    }
    printf("NULL");
}

struct Node* begin(struct Node* start)
{
    int data;
    struct Node *temp = start, *newNode = NULL;
    printf("\nEnter data for Node to be inserted in the Beginning: ");
    scanf("%i",&data);
    84:78

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu
≡ File Edit Search Run Compile Debug Project Options Window Help
[■] SAAD1.C 1-[↕]
newNode = createnode(data);
newNode->next = temp;
return newNode;
}

void search(struct Node* head)
{
    int data,count=1;
    struct Node *temp = head;
    printf("\nEnter data you want to search in the linked list: ");
    scanf("%i",&data);
    while(temp->data!=data)
    {
        temp = temp->next;
        count++;
    }
    printf("%i found in Node %i",data,count);
}
84:78

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu
```

Output: -

```
Enter the No. of Node: 3
Enter data for Node 1: 2
Enter data for Node 2: 3
Enter data for Node 3: 4
2 -> 3 -> 4 -> NULL
Enter data for Node to be inserted in the Beginning: 1
1 -> 2 -> 3 -> 4 -> NULL
Enter data you want to search in the linked list: 3
3 found in Node 3
```

**Practical Related Questions:**

1. Write a function to insert a node at the beginning of a Singly Linked List.

**Ans:**

```
struct Node* begin(struct Node* start)
{
    int data;
    struct Node *temp = start, *newNode = NULL;
    printf("\nEnter data for Node to be inserted in the Beginning: ");
    scanf("%i",&data);
    newNode = createnode(data);
    newNode->next = temp;
    return newNode;
}
```

2. Write a function to detect if a singly linked list has a cycle. If a cycle is detected, return the starting node of the cycle.

**Ans:**

```
void cyclecheck(struct Node* head)
{
    int flag = 0;
    struct Node *temp = NULL;
    temp=head->next;
    while(temp!=NULL)
    {
        if(temp==head)
        {
            printf("\nThere is a Cycle, Address of the first node of the cycle is %u",temp);
            flag = 1;
            break;
        }
        temp=temp->next;
    }
    if(flag==0)
    {
        printf("There is not cycle!");
    }
}
```

Marks Obtained			Dated signature of Teacher
Process Related (35)	Product Related (15)	Total (50)	