Page:

Page:
Date:

Experiment No. 3

Aim:- Implementation of Singly Linked list and various operations for real-world.

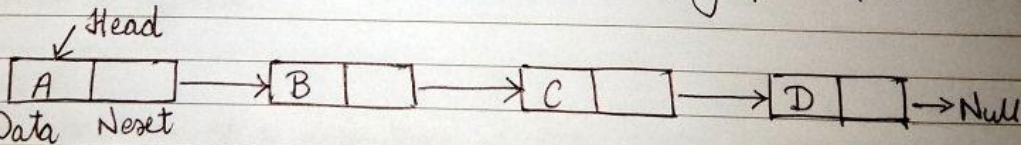
Objectives:-

1. To learn the basic principles of programming as applied to complex data structures
2. To learn the principales of linked list and its various operation.

Theory:-

Introduction To Linked Lists:

A linked list is a linear data structure, in which the elements are not stored at contiguous memory locations. The elements in a linked list are linked using pointers.



In simple words, a linked lists consists of nodes where each node contains a data field and a reference (link) to the next node in the list.

Singly Linked List: It is the simplest type of linked lists in which every node contains some data and a pointer to the next node of the same data type. The node contains a

Page:
Date:

Pointer to the next node means that the node stores the address of the next node in the sequence.

Insertion :-

The insertion into a singly linked list can be performed are different positions. Based on the position of the new node being inserted, the insertion is categorized into the following categories:

① Insertion at beginning → It involves inserting any element at the front of the list.
② Insertion at the end of the list → It involves insertion at the last of the linked list. The new node can be inserted as the only node in the list or it can be inserted as the last node.
③ Insertion after specified node. → It involves insertion after the specified node of the linked list. We need to skip the desired number of nodes in order to reach the node after which the new node will be inserted.

Deletion :-

The deletion of a node from a singly linked list can be performed at different position. Based on the position of node being deleted, the operation is categorized as:

① Deletion at beginning → It involves deletions of a node from the beginning of the list.
② Deletion at end → It involves deleting the last node of the list.
③ Deletion after specified Node → It involves deleting the node after the specified node in the list.

Page:
Date:

Traversing :-

In traversing, we simply visit each node of the list at least once in order to perform some specific operation on it.

Algorithm

Insertion in the beginning.

Step 1:- If PTR = NULL
Write overflow.
Go to Step 7
[End of If].
Step 2:- SET NEW_NODE = PTR.
Step 3:- SET PTR = PTR → NEXT.
Step 4:- SET NEW_NODE → DATA = VAL
Step 5:- SET NEW_NODE → NEXT = HEAD
Step 6:- SET HEAD = NEW_NODE
Step 7:- EXIT

Insertion at the End.

Step 1:- If PTR = NULL write overflow.
Go to Step 1
[End of If].
Step 2:- SET NEW_NODE = PTR.
Step 3:- SET PTR = PTR → NEXT.
Step 4:- SET NEW_NODE → DATA = VAL
Step 5:- SET NEW_NODE → NEXT = NULL
Step 6:- SET PTR = HEAD.
Step 7:- Repeat Step 8 while PTR → NEXT != NULL
Step 8:- SET PTR = PTR → NEXT

Page:

Page:
Date:

[End of Loop].
Step 9:- SET PTR → NEXT = NEW_NODE
Step 10:- EXIT

Insertion at specific node:
Step 1: If PTR = NULL
Write overflow
Go To Step 12
[End of If].
Step 2: SET NEW_NODE = PTR
Step 3: NEW_NODE → DATA = VAL
Step 4: SET TMP = HEAD
Step 5: SET 1 = 0
Step 6: Repeat Step 5 & 6 until 1
Step 7: TEMP = TEMP → NEXT
Step 8: If TEMP = NULL
Write "Desired Node Not Present".
Go to Step 12
End of If
End of LOOP
Step 9: PTR → NEXT = TEMP → NEXT
Step 10: TEMP → NEXT = PTR
Step 11: SET PTR = NEW_NODE
Step 12: EXIT

Page:
Date:

Deletion at beginning:

Step 1: If HEAD = NULL
Write underflow.
go to Step 5
[End of if]
Step 2: SET PTR = HEAD
Step 3: SET HEAD = HEAD → NEXT
Step 4: FREE PTR
Step 5: EXIT

Deletion at Specified Node.

Step 1: If HEAD = NULL
Write underflow.
Goto Step 10
END of IF
Step 2: SET TEMP = HEAD
Step 3: SET I = 0
Step 4: Repeat Step 5 to 8 until I
Step 5: TEMP1 = TEMP
Step 6: TEMP = TEMP → NEXT
Step 7: If TEMP = NULL
Write "Desired Node Not Present"
Goto Step 12.
End of If
Step 8: I = I + 1
End of Loop.
Step 9: TEMP1 → NEXT = TEMP → NEXT
Step 10: FREE . TEMP.
Step 11: EXIT.

Page:
Date:

Deletion at the End.

Step 1 : If HEAD = NULL
Write underflow.
Go to Step 8
[End of If]
Step 2 : SET PTR = HEAD
Step 3 : Repeat Steps 4 & 5 while PTR → NEXT != NULL
Step 4 : SET PREPTR = PTR
Step 5 : SET PTR = PTR → NEXT
[End of Loop]
Step 6 : SET PREPTR → NEXT = NULL
Step 7 : FREE PTR.
Step 8 : EXIT.

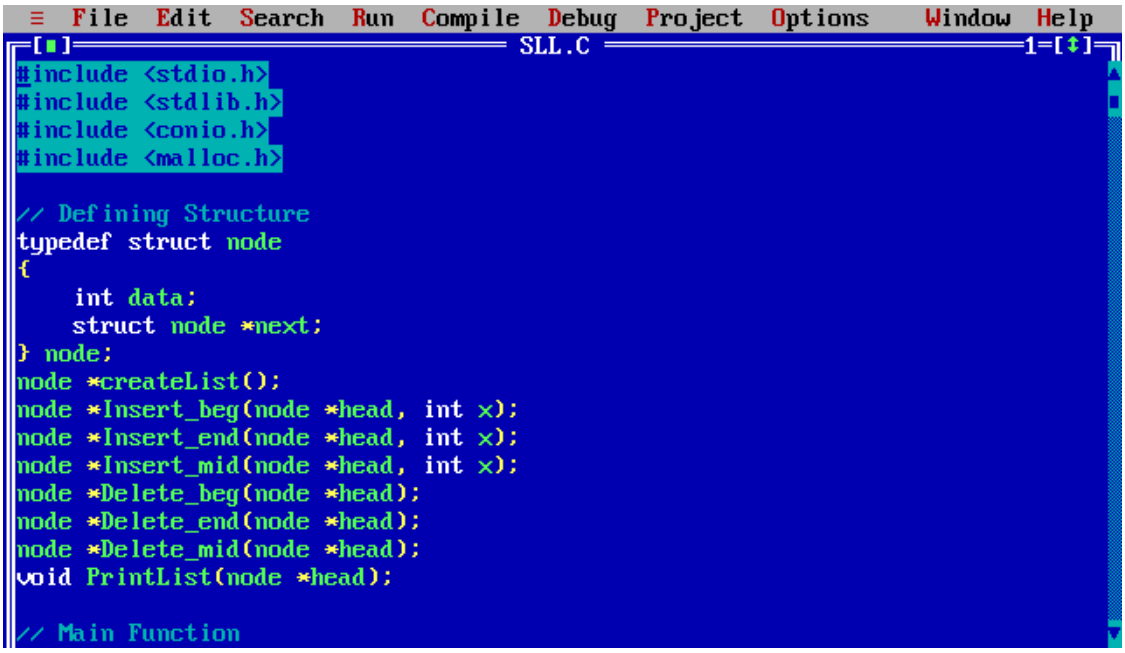
Example :-

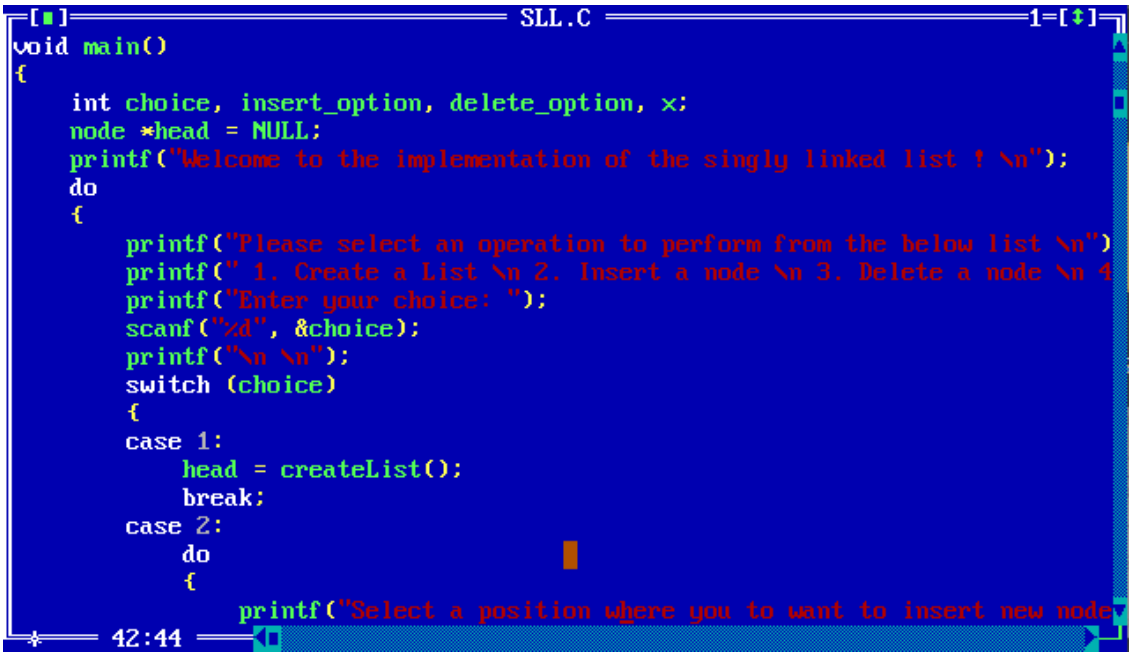
1. List of images that need to be burned to a CD in a medical imaging application.
2. List of objects in a 3D game that need to be emailed some notification.
3. List of users of a website that need to be rendered to the screen.

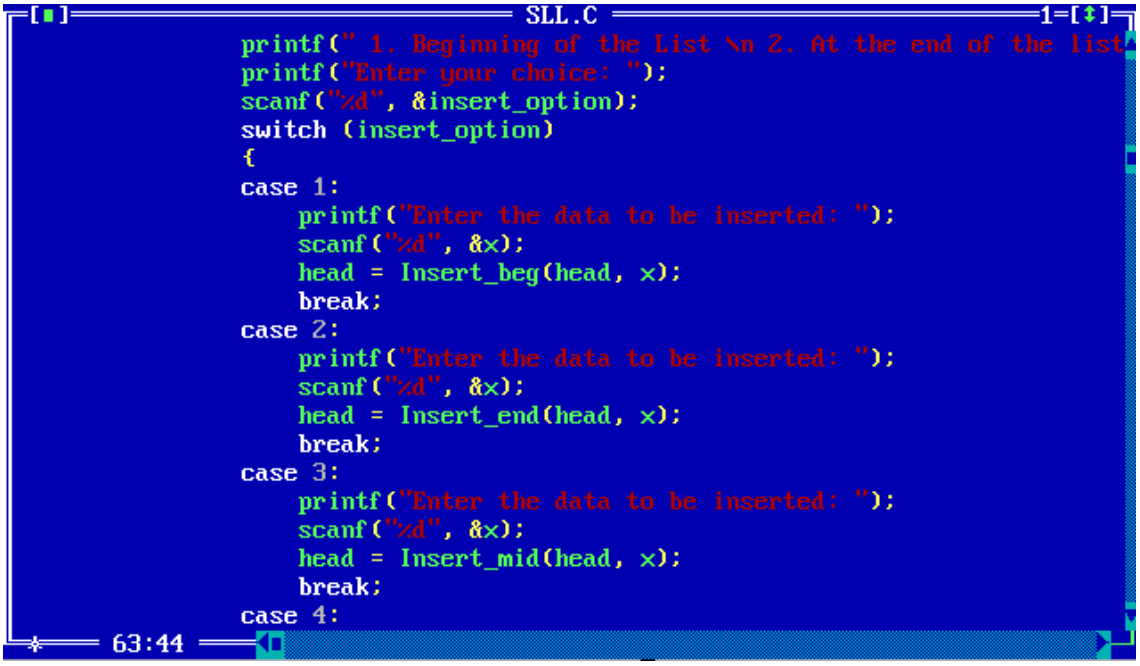
Conclusion :- Thus, we have studied the concepts & implementation of singly linked list and its various operations.

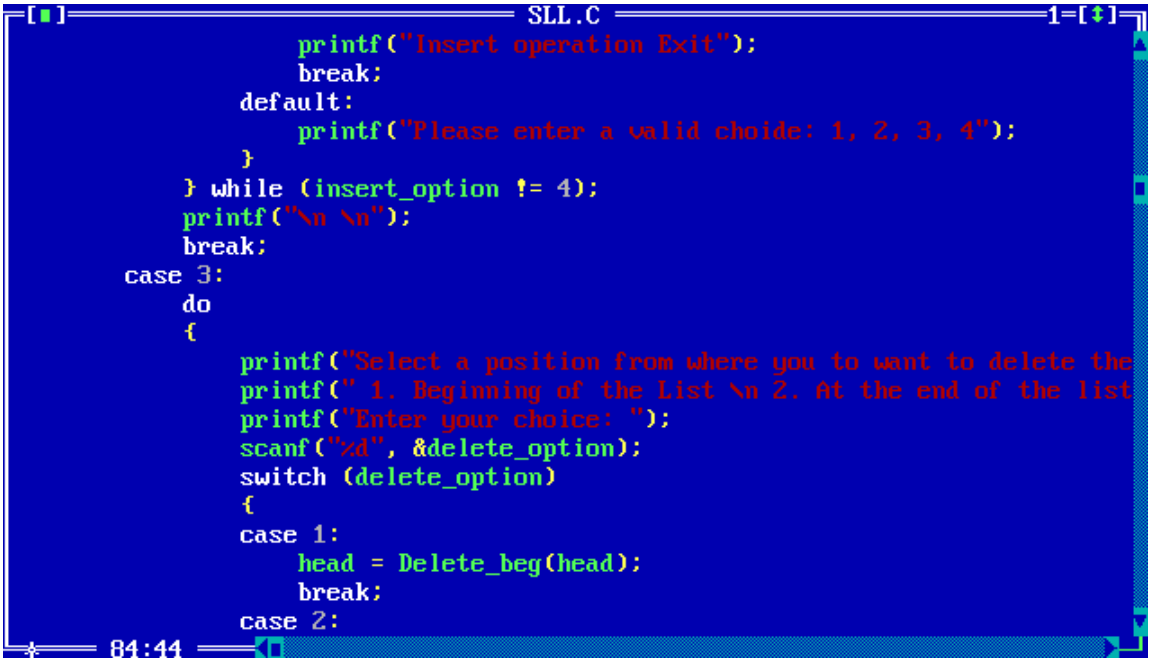
Outcome :- Apply the concepts of singly and doubly linked list for real-world application.

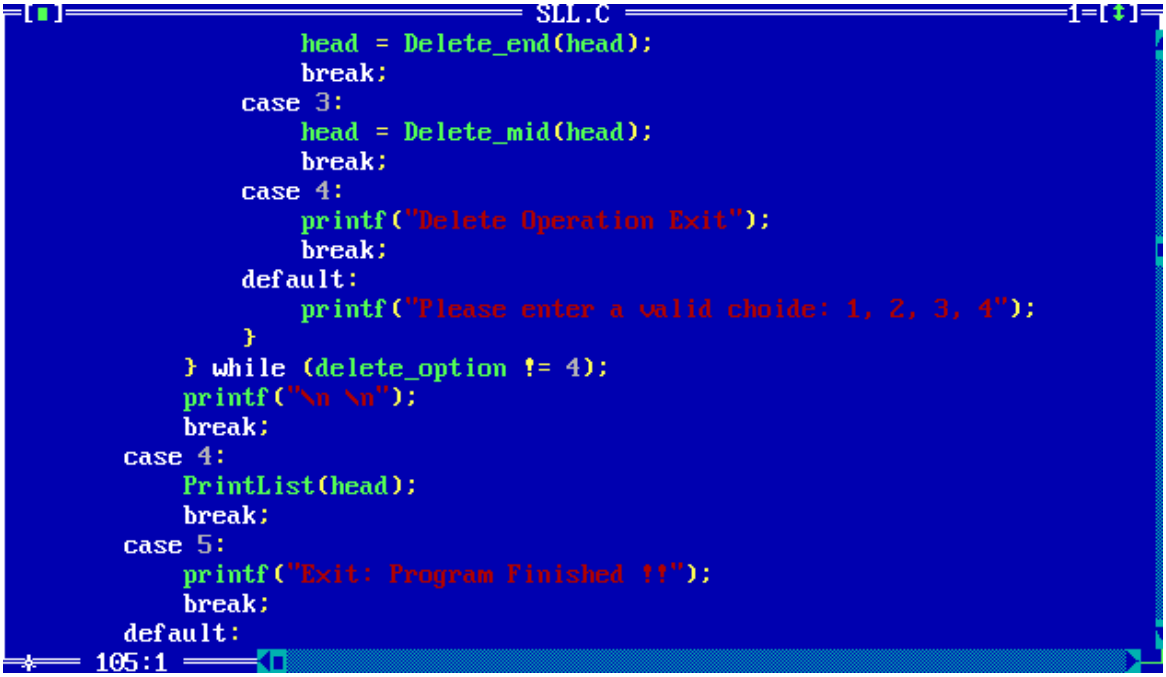
Program: SLL

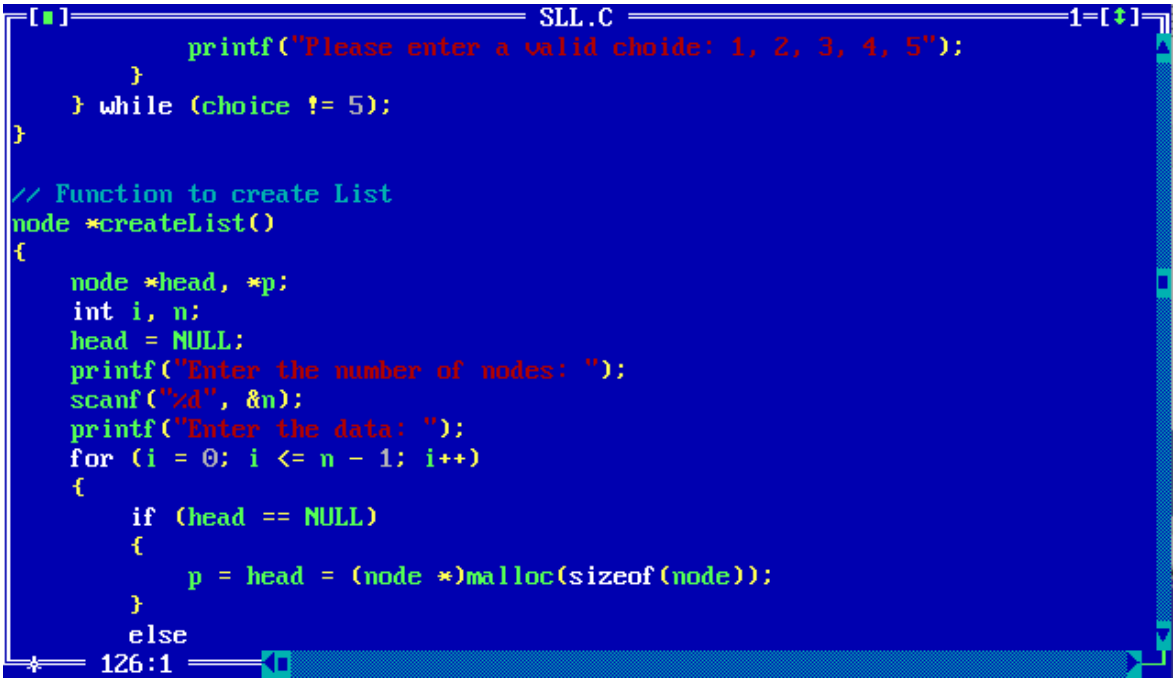

File Edit Search Run Compile Debug Project Options Window Help
SLL.C
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <malloc.h>
// Defining Structure
typedef struct node
{
int data;
struct node *next;
} node;
node *createList();
node *Insert_beg(node *head, int x);
node *Insert_end(node *head, int x);
node *Insert_mid(node *head, int x);
node *Delete_beg(node *head);
node *Delete_end(node *head);
node *Delete_mid(node *head);
void PrintList(node *head);
// Main Function

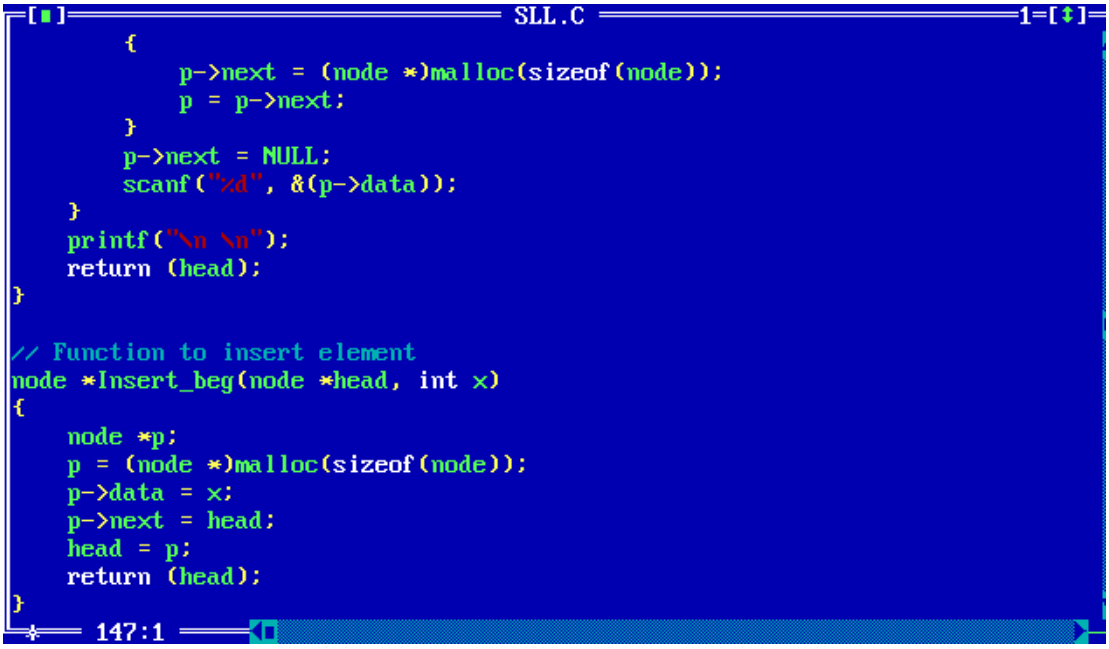

SLL.C
void main()
{
int choice, insert_option, delete_option, x;
node *head = NULL;
printf("Welcome to the implementation of the singly linked list ! \n");
do
{
printf("Please select an operation to perform from the below list \n")
printf(" 1. Create a List \n 2. Insert a node \n 3. Delete a node \n 4
printf("Enter your choice: ");
scanf("%d", &choice);
printf("\n \n");
switch (choice)
{
case 1:
head = createList();
break;
case 2:
do
{
printf("Select a position where you to want to insert new node
42:44

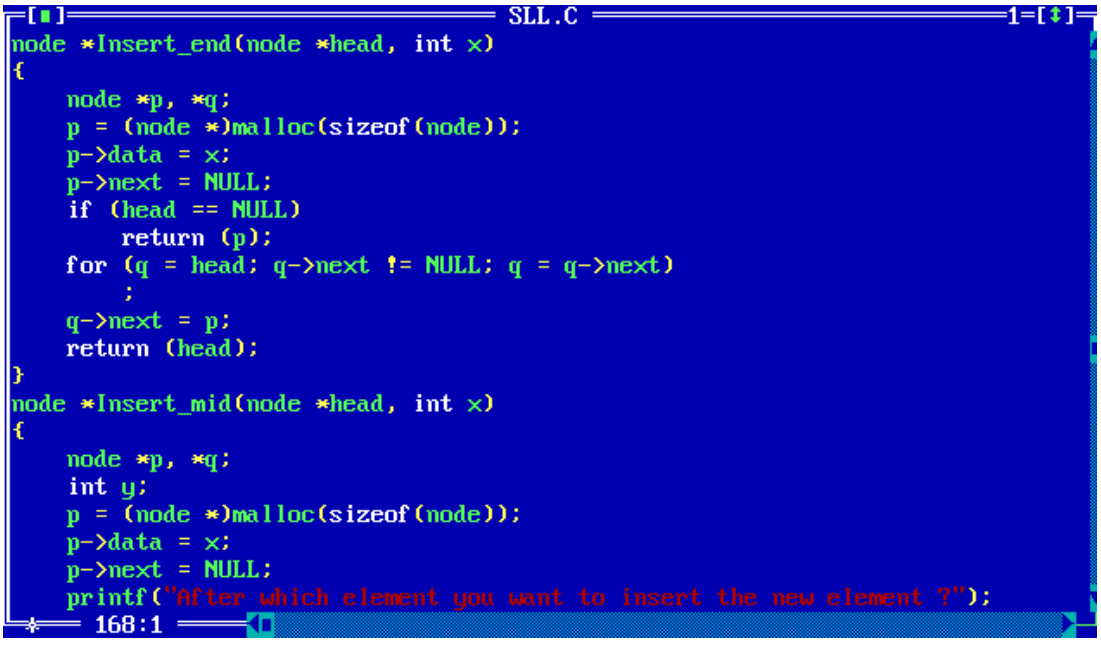

SLL.C
printf(" 1. Beginning of the List \n 2. At the end of the list
printf("Enter your choice: ");
scanf("%d", &insert_option);
switch (insert_option)
{
case 1:
printf("Enter the data to be inserted: ");
scanf("%d", &x);
head = Insert_beg(head, x);
break;
case 2:
printf("Enter the data to be inserted: ");
scanf("%d", &x);
head = Insert_end(head, x);
break;
case 3:
printf("Enter the data to be inserted: ");
scanf("%d", &x);
head = Insert_mid(head, x);
break;
case 4:
63:44

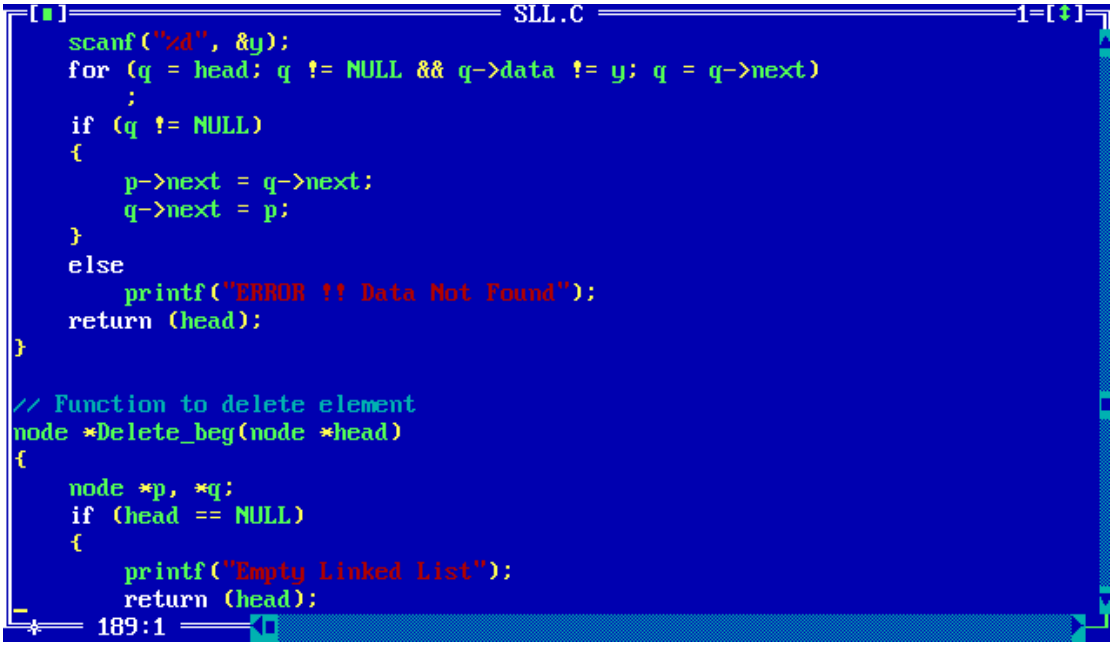

SLL.C
printf("Insert operation Exit");
break;
default:
printf("Please enter a valid choide: 1, 2, 3, 4");
}
} while (insert_option != 4);
printf("\n \n");
break;
case 3:
do
{
printf("Select a position from where you to want to delete the
printf(" 1. Beginning of the List \n 2. At the end of the list
printf("Enter your choice: ");
scanf("%d", &delete_option);
switch (delete_option)
{
case 1:
head = Delete_beg(head);
break;
case 2:
84:44

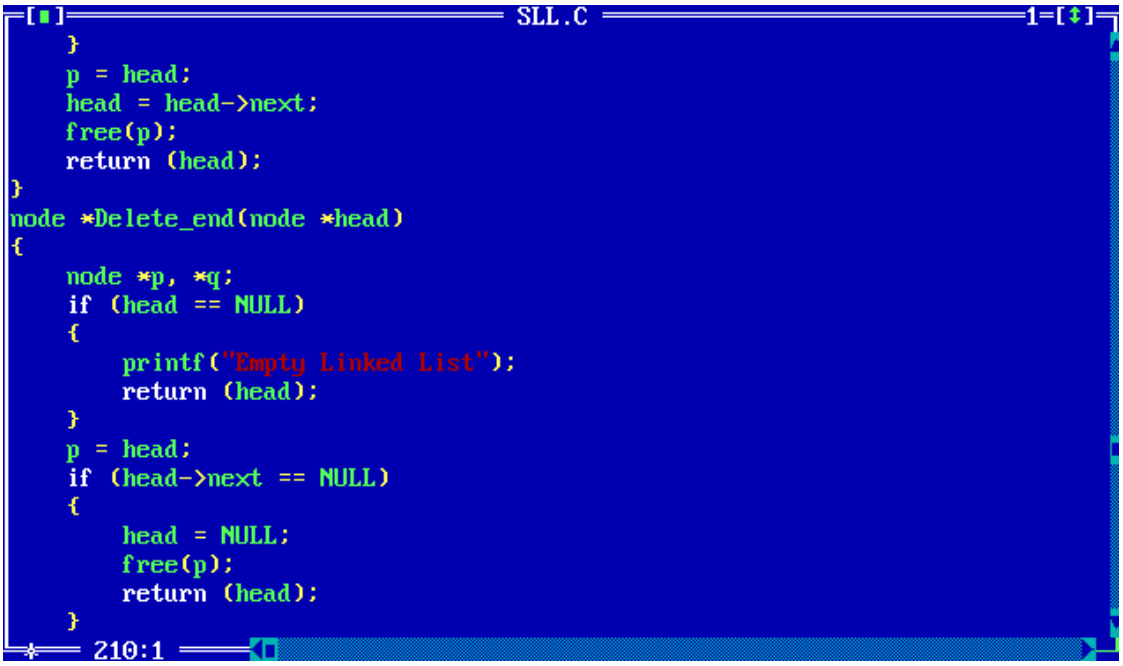

SLL.C
head = Delete_end(head);
break;
case 3:
head = Delete_mid(head);
break;
case 4:
printf("Delete Operation Exit");
break;
default:
printf("Please enter a valid choide: 1, 2, 3, 4");
} while (delete_option != 4);
printf("\n \n");
break;
case 4:
PrintList(head);
break;
case 5:
printf("Exit: Program Finished !!");
break;
default:
105:1

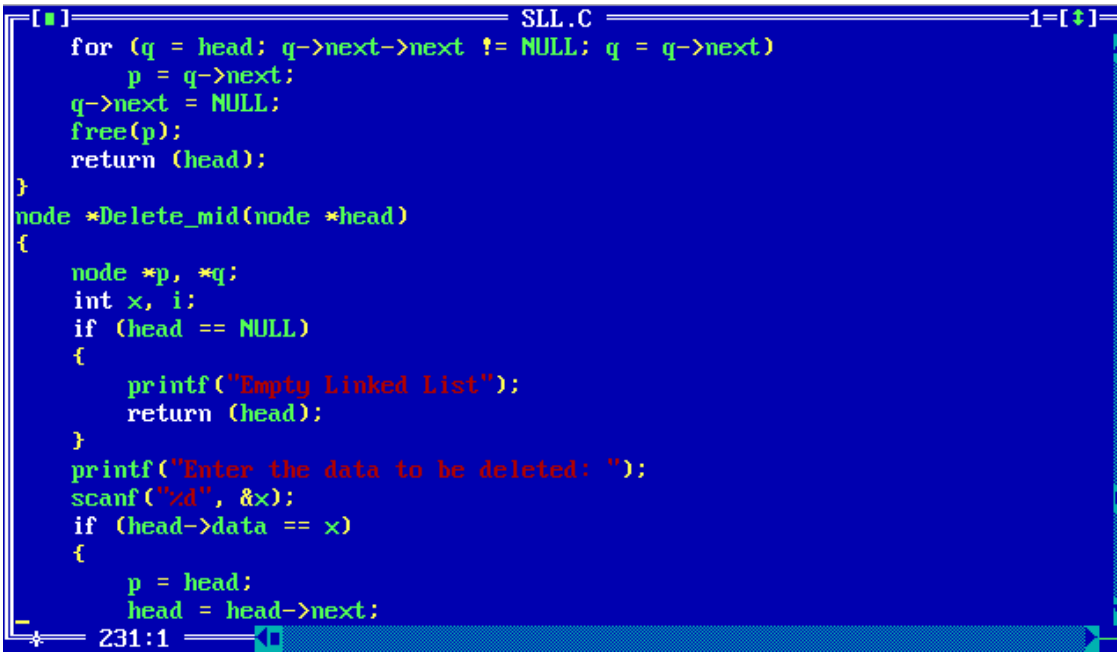

SLL.C
printf("Please enter a valid choide: 1, 2, 3, 4, 5");
}
} while (choice != 5);
}
// Function to create List
node *createList()
{
node *head, *p;
int i, n;
head = NULL;
printf("Enter the number of nodes: ");
scanf("%d", &n);
printf("Enter the data: ");
for (i = 0; i <= n - 1; i++)
{
if (head == NULL)
{
p = head = (node *)malloc(sizeof(node));
}
else
126:1

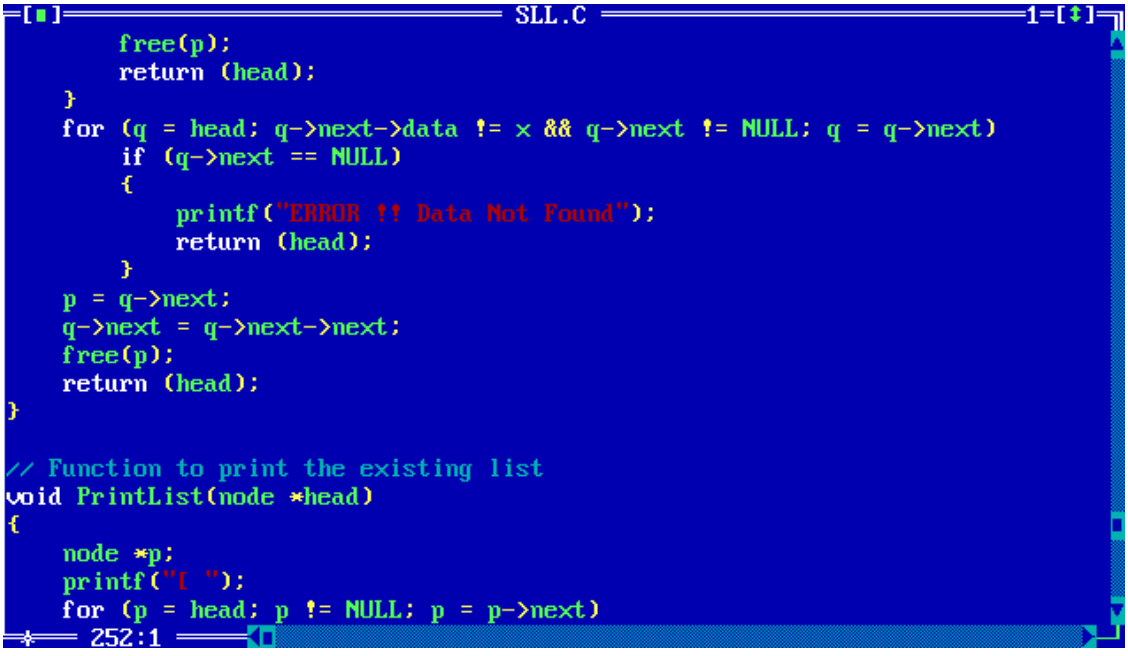

SLL.C
{
p->next = (node *)malloc(sizeof(node));
p = p->next;
}
p->next = NULL;
scanf("%d", &(p->data));
}
printf("\n \n");
return (head);
}
// Function to insert element
node *Insert_beg(node *head, int x)
{
node *p;
p = (node *)malloc(sizeof(node));
p->data = x;
p->next = head;
head = p;
return (head);
}
147:1

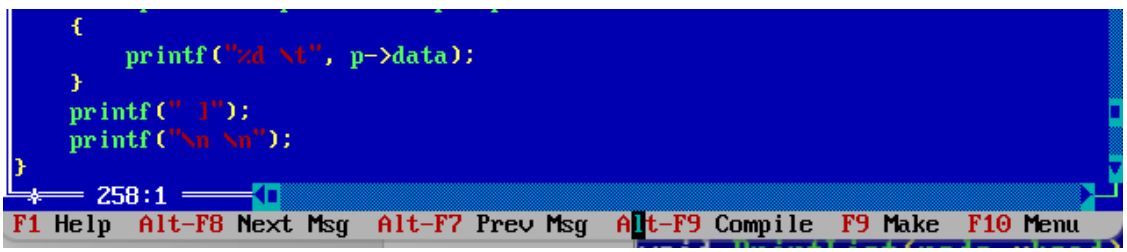

SLL.C
node *Insert_end(node *head, int x)
{
node *p, *q;
p = (node *)malloc(sizeof(node));
p->data = x;
p->next = NULL;
if (head == NULL)
return (p);
for (q = head; q->next != NULL; q = q->next)
;
q->next = p;
return (head);
}
node *Insert_mid(node *head, int x)
{
node *p, *q;
int y;
p = (node *)malloc(sizeof(node));
p->data = x;
p->next = NULL;
printf("After which element you want to insert the new element ?");
168:1


SLL.C
1=[↕]
scanf("%d", &y);
for (q = head; q != NULL && q->data != y; q = q->next)
;
if (q != NULL)
{
p->next = q->next;
q->next = p;
}
else
printf("ERROR !! Data Not Found");
return (head);
}
// Function to delete element
node *Delete_beg(node *head)
{
node *p, *q;
if (head == NULL)
{
printf("Empty Linked List");
return (head);
189:1


SLL.C
1=[↕]
}
p = head;
head = head->next;
free(p);
return (head);
}
node *Delete_end(node *head)
{
node *p, *q;
if (head == NULL)
{
printf("Empty Linked List");
return (head);
}
p = head;
if (head->next == NULL)
{
head = NULL;
free(p);
return (head);
}
210:1


SLL.C
1=[↕]
for (q = head; q->next->next != NULL; q = q->next)
p = q->next;
q->next = NULL;
free(p);
return (head);
}
node *Delete_mid(node *head)
{
node *p, *q;
int x, i;
if (head == NULL)
{
printf("Empty Linked List");
return (head);
}
printf("Enter the data to be deleted: ");
scanf("%d", &x);
if (head->data == x)
{
p = head;
head = head->next;
231:1


SLL.C
1=[↕]
free(p);
return (head);
}
for (q = head; q->next->data != x && q->next != NULL; q = q->next)
if (q->next == NULL)
{
printf("ERROR !! Data Not Found");
return (head);
}
p = q->next;
q->next = q->next->next;
free(p);
return (head);
}
// Function to print the existing list
void PrintList(node *head)
{
node *p;
printf("[");
for (p = head; p != NULL; p = p->next)
252:1


{
printf("%d \t", p->data);
}
printf("]");
printf("\n \n");
}
258:1
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

OUTPUT:-

1. Value Inserted[10,20,30,40,50]

```
Welcome to the implementation of the singly linked list !
Please select an operation to perform from the below list
 1. Create a List
 2. Insert a node
 3. Delete a node
 4. Print the existing list
 5. Exit
Enter your choice: 1


Enter the number of nodes: 5
Enter the data: 10
20
30
40
50


Please select an operation to perform from the below list
 1. Create a List
 2. Insert a node
 3. Delete a node
 4. Print the existing list
 5. Exit
Enter your choice:
```

2. Display of Singly Linked Links

```
Enter the data: 10
20
30
40
50


Please select an operation to perform from the below list
 1. Create a List
 2. Insert a node
 3. Delete a node
 4. Print the existing list
 5. Exit
Enter your choice: 4


[ 10    20      30      40      50       ]

Please select an operation to perform from the below list
 1. Create a List
 2. Insert a node
 3. Delete a node
 4. Print the existing list
 5. Exit
Enter your choice:
```

3. Inserted Node

- Begnning Of the List

```
 3. Insert in between
 4. Exit the insert operation
Enter your choice: 1
Enter the data to be inserted: 11
Select a position where you to want to insert new node
 1. Beginning of the List
 2. At the end of the list
 3. Insert in between
 4. Exit the insert operation
Enter your choice: 2
Enter the data to be inserted: 12
Select a position where you to want to insert new node
 1. Beginning of the List
 2. At the end of the list
 3. Insert in between
 4. Exit the insert operation
Enter your choice:      3
Enter the data to be inserted: 13
After which element you want to insert the new element ?20
Select a position where you to want to insert new node
 1. Beginning of the List
 2. At the end of the list
 3. Insert in between
 4. Exit the insert operation
Enter your choice:
```

- Ending Of the List/ Inserted in between.

```
[ 10    20      30      40      50       ]

Please select an operation to perform from the below list
 1. Create a List
 2. Insert a node
 3. Delete a node
 4. Print the existing list
 5. Exit
Enter your choice:      2


Select a position where you to want to insert new node
 1. Beginning of the List
 2. At the end of the list
 3. Insert in between
 4. Exit the insert operation
Enter your choice: 1
Enter the data to be inserted: 11
Select a position where you to want to insert new node
 1. Beginning of the List
 2. At the end of the list
 3. Insert in between
 4. Exit the insert operation
Enter your choice:
```

- Display Of Inserted List.

```
 1. Beginning of the List
 2. At the end of the list
 3. Insert in between
 4. Exit the insert operation
Enter your choice:      4
Insert operation Exit

Please select an operation to perform from the below list
 1. Create a List
 2. Insert a node
 3. Delete a node
 4. Print the existing list
 5. Exit
Enter your choice: 4


[ 11    10      20      13      30      40      50      12      ]

Please select an operation to perform from the below list
 1. Create a List
 2. Insert a node
 3. Delete a node
 4. Print the existing list
 5. Exit
Enter your choice: _
```

4. Deletion Node.

- Beginning Of the Lists

```
[ 11    10      20      13      30      40      50      12      ]

Please select an operation to perform from the below list
 1. Create a List
 2. Insert a node
 3. Delete a node
 4. Print the existing list
 5. Exit
Enter your choice: 3


Select a position from where you to want to delete the element
 1. Beginning of the List
 2. At the end of the list
 3. Somewhere in between
 4. Exit the delete operation
Enter your choice: 1
Select a position from where you to want to delete the element
 1. Beginning of the List
 2. At the end of the list
 3. Somewhere in between
 4. Exit the delete operation
Enter your choice:
```

- Delection At end.

```
Select a position from where you to want to delete the element
 1. Beginning of the List
 2. At the end of the list
 3. Somewhere in between
 4. Exit the delete operation
Enter your choice: 1
Select a position from where you to want to delete the element
 1. Beginning of the List
 2. At the end of the list
 3. Somewhere in between
 4. Exit the delete operation
Enter your choice: 2
Select a position from where you to want to delete the element
 1. Beginning of the List
 2. At the end of the list
 3. Somewhere in between
 4. Exit the delete operation
Enter your choice: 3
Enter the data to be deleted: 30
Select a position from where you to want to delete the element
 1. Beginning of the List
 2. At the end of the list
 3. Somewhere in between
 4. Exit the delete operation
Enter your choice:
```

- Delection Somewhere in between

```
Enter your choice: 3


Select a position from where you to want to delete the element
 1. Beginning of the List
 2. At the end of the list
 3. Somewhere in between
 4. Exit the delete operation
Enter your choice: 3
Enter the data to be deleted: 30
Select a position from where you to want to delete the element
 1. Beginning of the List
 2. At the end of the list
 3. Somewhere in between
 4. Exit the delete operation
Enter your choice: 4
Delete Operation Exit

Please select an operation to perform from the below list
 1. Create a List
 2. Insert a node
 3. Delete a node
 4. Print the existing list
 5. Exit
Enter your choice:
```

5. Display The List After the operation performed