Batch: B1 **Experiment Number: 2**

Roll Number: 16010420133 and 16010420141 **Name: Soumen Samanta and Omkar Karbhari**

Aim of the Experiment: Implementing Singly Linked List (SLL) supporting following operations using menu driven program.

1. Insert at the Begin
2. Insert after the specified existing node
3. Delete before the specified existing node
4. Display all elements

Resources Used: Turbo C/ C++ editor and compiler (online or offline).

Algorithm:

Program should implement the specified operations strictly in the following manner. Also implement a support method isempty() and make use of it at appropriate places.

1. **createSLL()** – This void function should create a START/HEAD pointer with NULL value as empty SLL.
2. **insertBegin(typedef newelement)** – This void function should take a newelement as an argument to be inserted on an existing SLL and insert it before the element pointed by the START/HEAD pointer.
3. **insertAfter(typedef newelement, typedef existingelement)** – This void function should take two arguments. The function should search for an existingelement on non-empty SLL and insert newelement after this element.
4. **typedef deleteBefore(typedef existingelement)** – This function should search for the existing element passed to the function in the non-empty SLL, delete the node siting before it and return the deleted element.
5. **display( )** – This is a void function which should go through non- empty SLL starting from START/HEAD pointer and display each element of the SLL till the end.

Program/ Steps:

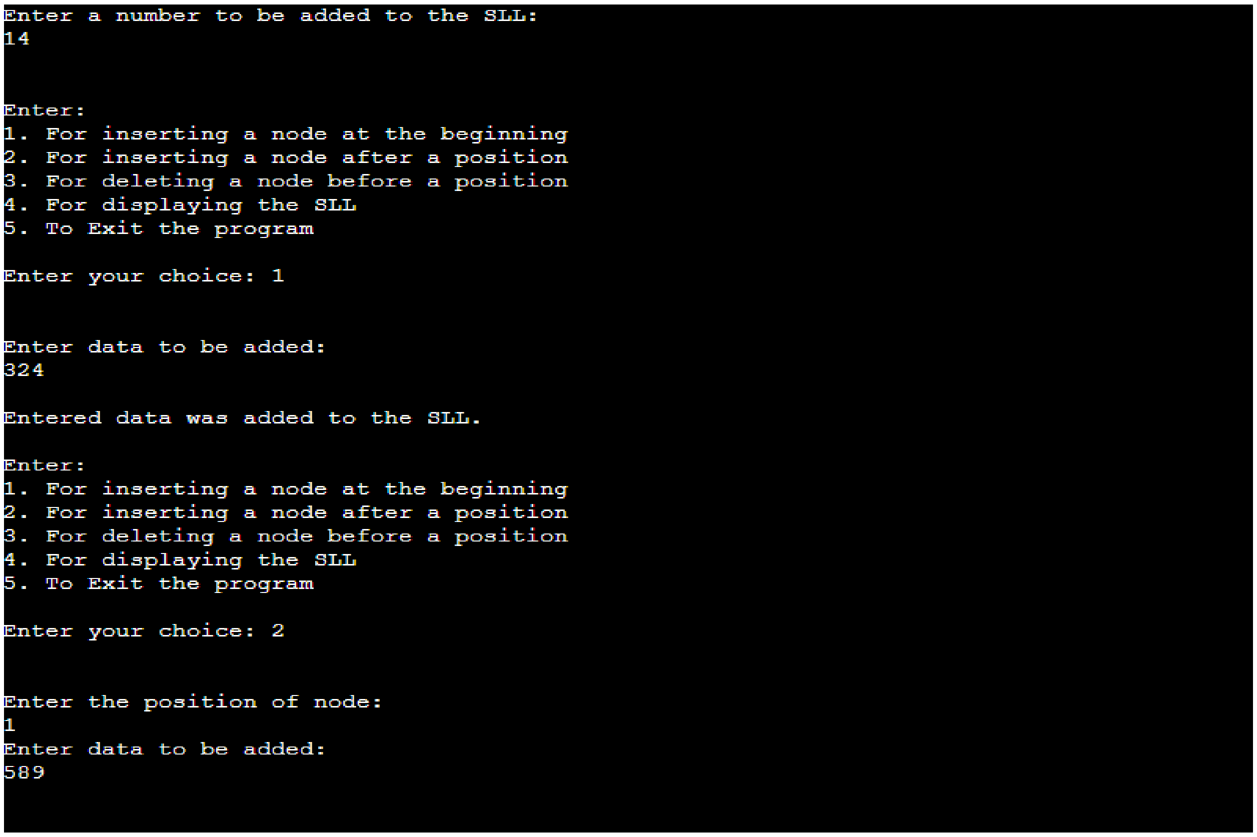
```
#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
#include <string.h>
struct node
{
   struct node *next;
   int data;
};
struct node *head, *start = NULL, *begin = NULL;
```

```
void insertBegin(int d)
{
struct node *newnode;
   newnode = (struct node *)malloc(sizeof(struct node));
   newnode->next = head;
   newnode->data = d;
   head = newnode;
}
void insertAfter(int pos, int d)
{
   struct node *newnode;
   newnode = (struct node *)malloc(sizeof(struct node));
   struct node *t1, *t2;
   t1 = (struct node *)malloc(sizeof(struct node));
   t2 = (struct node *)malloc(sizeof(struct node));
   newnode->data = d;
   newnode->next = NULL;
   int c = 0;
   if (pos == 0)
   {
      insertBegin(d);
   }
   else
   {
      t1 = head;
      while (c < pos && t1->next != NULL)
      {
         t2 = t1;
         t1 = t1->next;
         c++;
      }
      if (c == pos)
      {
         newnode->next = t1;
         t2->next = newnode;
      }
      else if (c == pos - 1)
      {
         t1->next = newnode;
      }
      else
      {
         printf("The given node does not exist.");
      }
```

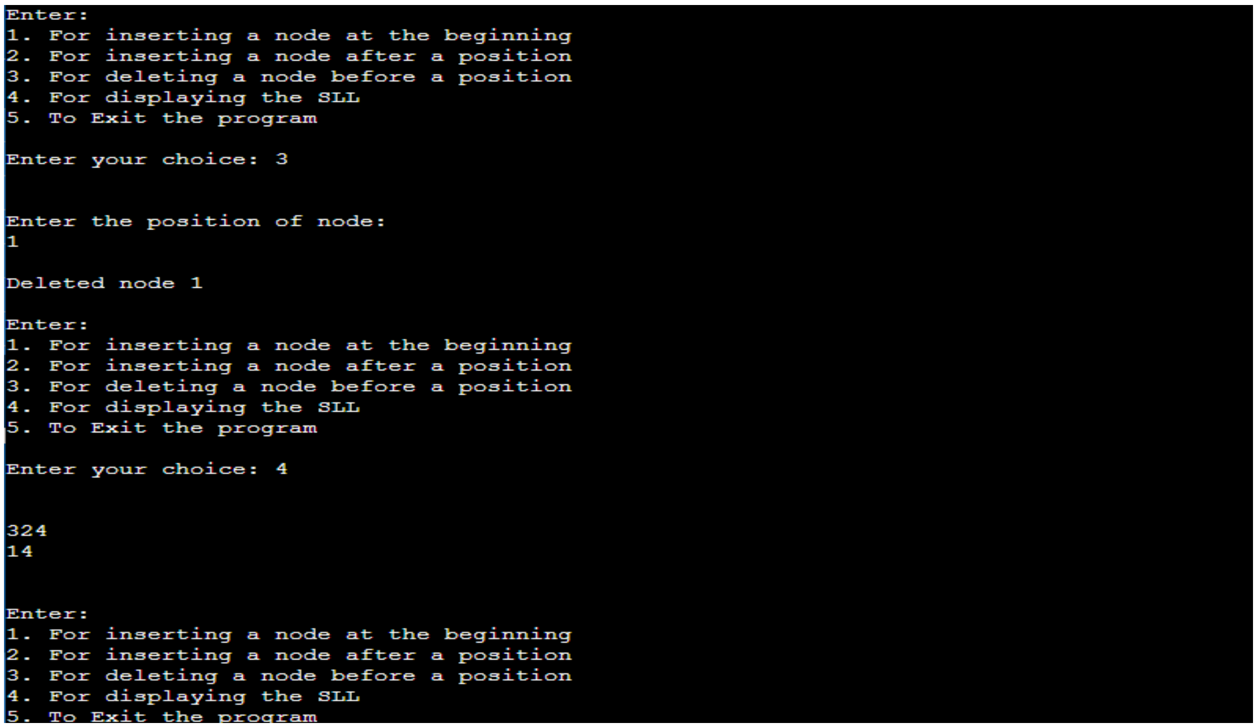
```
    }
}
void deleteBefore()
{
    struct node *ptr, *ptr1;
    int loc, i;
    scanf("%d", &loc);
    ptr = head;
    for (i = 0; i < loc; i++)
    {
        ptr1 = ptr;
        ptr = ptr->next;
        if (ptr == NULL)
        {
            printf("\nThere are less than %d elements in the list..\n", loc);
            return;
        }
    }

    ptr1->next = ptr->next;
    free(ptr);
    printf("\nDeleted node %d", loc);
}
void display()
{
    begin = head;
    while (begin != NULL)
    {
        printf("%d\n", begin->data);
        begin = begin->next;
    }
}
int main()
{
    int n, d1, p, d;
    start = (struct node *)malloc(sizeof(struct node));
    puts("Enter a number to be added to the SLL: ");
    scanf("%d", &d1);
    start->data = d1;
    start->next = head;
    head = start;
    while (1)
    {
        puts("\n");
    puts("Enter:\n1. For inserting a node at the beginning\n2. For inserting a node after a
position\n3. For deleting a node before a position\n4. For displaying the SLL\n5. To Exit the
program");
    printf("\nEnter your choice: ");
    scanf("%d", &n);
    puts("\n");
    switch (n)
    {
    case 1:
    {
      puts("Enter data to be added:");
      scanf("%d", &d);
      insertBegin(d);
      printf("\nEntered data was added to the SLL.");
      break;
    }
    case 2:
    {
      puts("Enter the position of node:");
      scanf("%d",&p);
      puts("Enter data to be added:");
      scanf("%d", &d);
      insertAfter(p, d);
      break;
    }
    case 3:
    {
      printf("Enter the position of node:\n");
      deleteBefore();
      break;
    }
    case 4:
    {
      display();
      break;
    }
    case 5:
    {
      printf("\nThe program is being exited...\n");
      exit(1);
    }
    default:
    {
      puts("Please enter one of the given options.");
```

```
         break;
       }
       }
    }
}
```

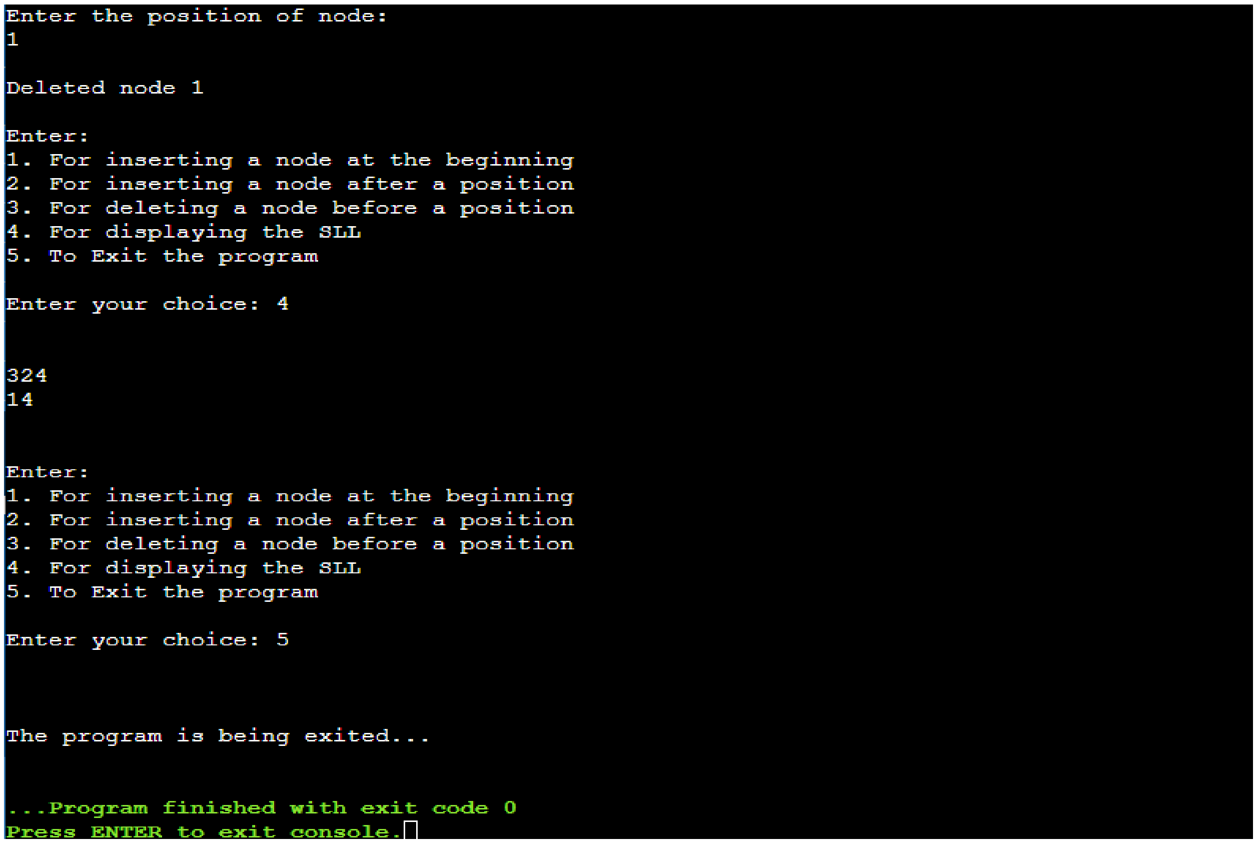
Output/Result:



```
Enter a number to be added to the SLL:
14


Enter:
1. For inserting a node at the beginning
2. For inserting a node after a position
3. For deleting a node before a position
4. For displaying the SLL
5. To Exit the program

Enter your choice: 1


Enter data to be added:
324

Entered data was added to the SLL.

Enter:
1. For inserting a node at the beginning
2. For inserting a node after a position
3. For deleting a node before a position
4. For displaying the SLL
5. To Exit the program

Enter your choice: 2


Enter the position of node:
1
Enter data to be added:
589
```



```
Enter:
1. For inserting a node at the beginning
2. For inserting a node after a position
3. For deleting a node before a position
4. For displaying the SLL
5. To Exit the program

Enter your choice: 3


Enter the position of node:
1

Deleted node 1

Enter:
1. For inserting a node at the beginning
2. For inserting a node after a position
3. For deleting a node before a position
4. For displaying the SLL
5. To Exit the program

Enter your choice: 4


324
14


Enter:
1. For inserting a node at the beginning
2. For inserting a node after a position
3. For deleting a node before a position
4. For displaying the SLL
5. To Exit the program
```



```
Enter the position of node:
1

Deleted node 1

Enter:
1. For inserting a node at the beginning
2. For inserting a node after a position
3. For deleting a node before a position
4. For displaying the SLL
5. To Exit the program

Enter your choice: 4


324
14


Enter:
1. For inserting a node at the beginning
2. For inserting a node after a position
3. For deleting a node before a position
4. For displaying the SLL
5. To Exit the program

Enter your choice: 5



The program is being exited...


...Program finished with exit code 0
Press ENTER to exit console.
```

Conclusion (based on the Results and outcomes achieved):

From the above experiment I was successfully able to implement the usage of linked list in C language. I got familiar with using pointers to structures, creating nodes, assigning values using pointers, traversing through the linked list using pointers and deleting elements in the linked list. Linked lists are beneficial to save memory space much more than arrays.

References:

Books/ Journals/ Websites:

- Y. Langsam, M. Augenstin and A. Tannenbaum, "Data Structures using C", Pearson Education Asia, 1st Edition, 2002.
- E. Horowitz, S. Sahni, S.Anderson-freed, "Fundamentals of Data Structures in C", 2nd Edition, University Press