DSA Assignment: 5

Exp 5: Implementation of Singly Linked List

Shashwat Tripathi

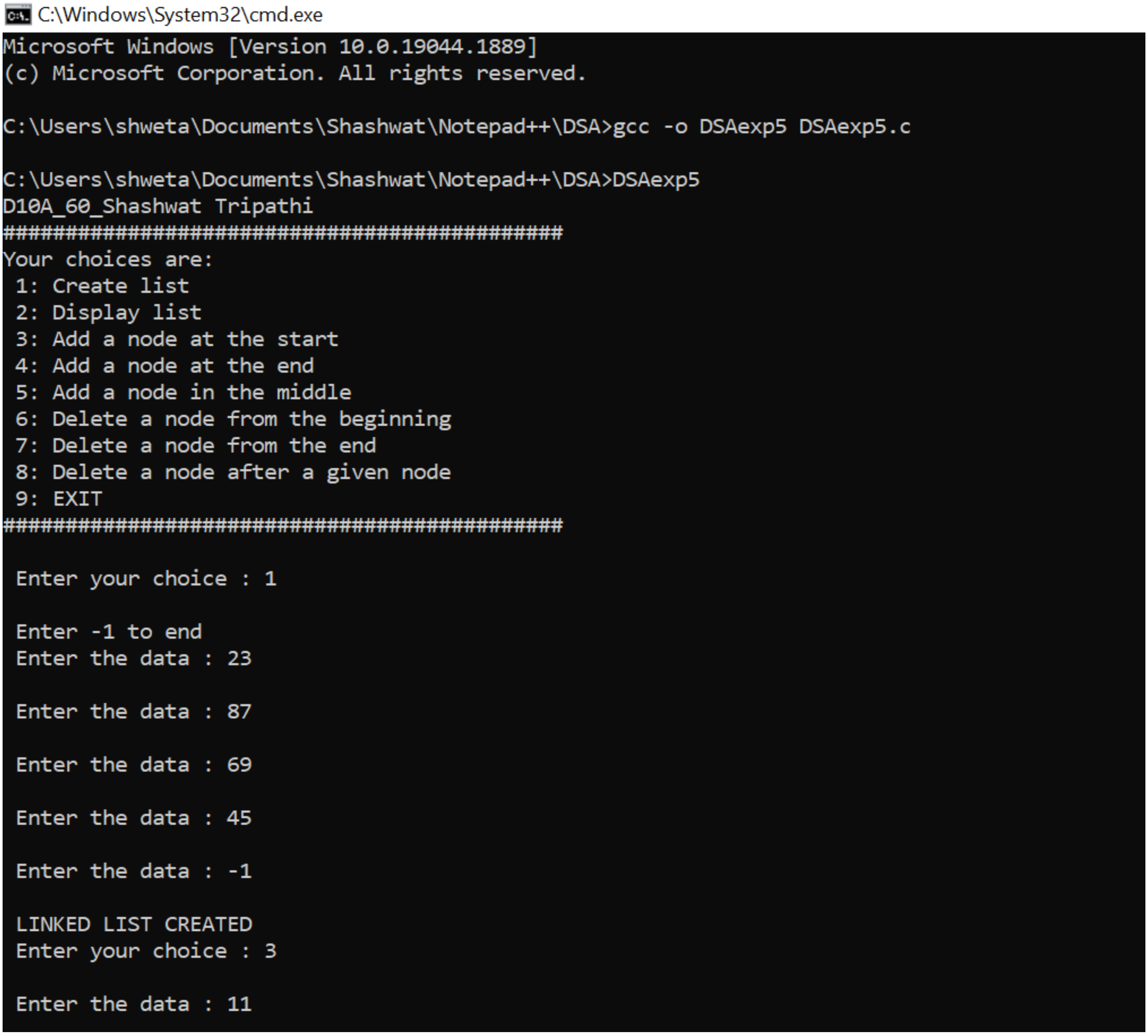
D10A Roll No: 60

AIM: In this experiment, we will implement Singly Linked List.

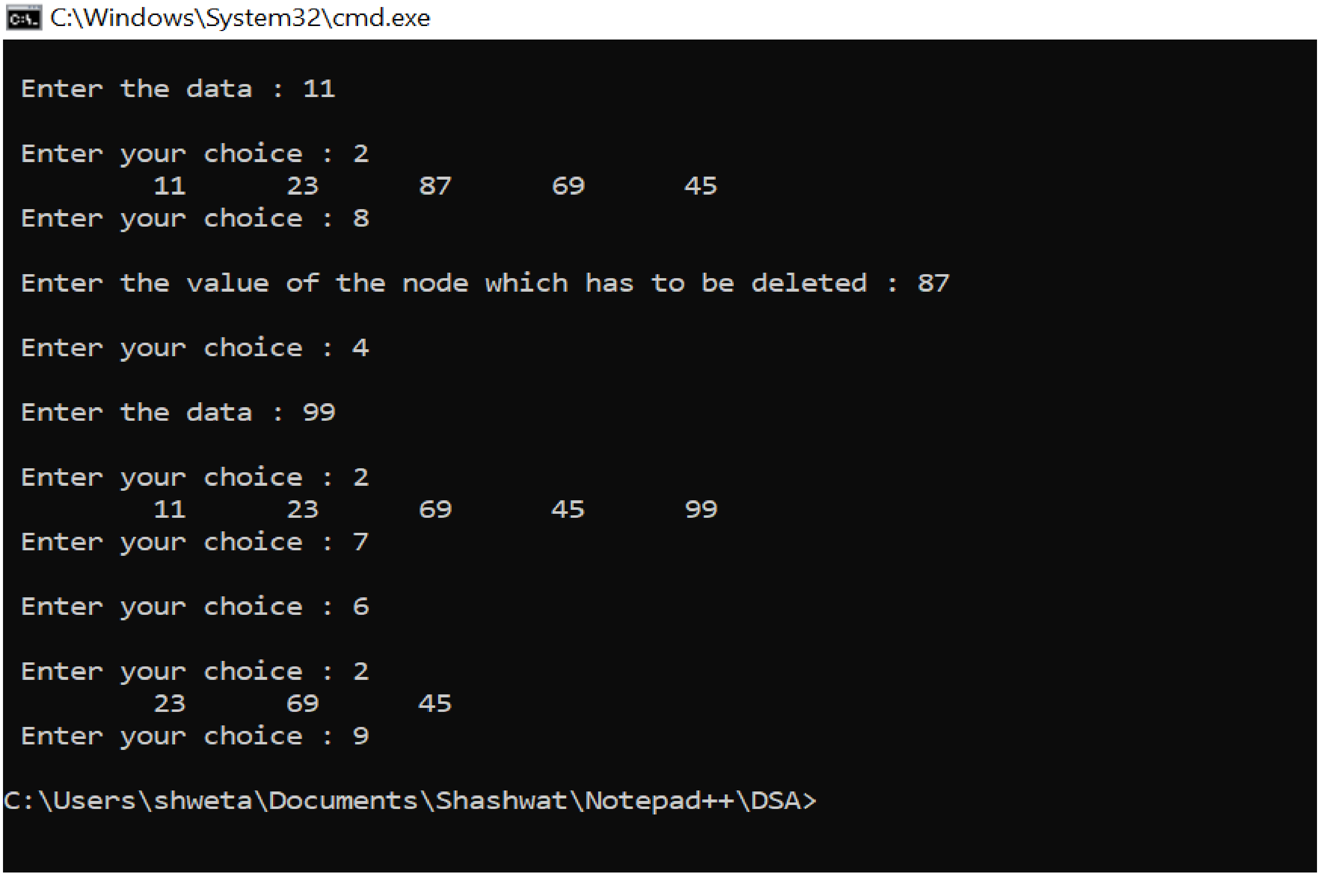
CODE:

```
// Exp 05 Implementation of Singly Linked List.
#include <stdio.h>
#include <stdlib.h>

struct node
{
    int data;
    struct node *next;
};
struct node *start = NULL;
struct node *create(struct node *);
struct node *display(struct node *);
struct node *insertbeginning(struct node *);
struct node *insertend(struct node *);
struct node *insertmiddle(struct node *);
struct node *deletebeginning(struct node *);
struct node *deleteend(struct node *);
struct node *deletemiddle(struct node *);

int main(int argc, char *argv[])
{
    int choice;
    printf("D10A_60_Shashwat Tripathi");
    printf("\n#############################################\n");
    printf("Your choices are: ");
        printf("\n 1: Create list");
        printf("\n 2: Display list");
        printf("\n 3: Add a node at the start");
        printf("\n 4: Add a node at the end");
        printf("\n 5: Add a node in the middle");
        printf("\n 6: Delete a node from the beginning");
        printf("\n 7: Delete a node from the end");
        printf("\n 8: Delete a node after a given node");
        printf("\n 9: EXIT");
    printf("\n#############################################\n");

    do
    {

        printf("\n Enter your choice : ");
        scanf("%d", &choice);
        switch (choice)
        {
            case 1:
                start = create(start);
                printf("\n LINKED LIST CREATED");
                break;
            case 2:
                start = display(start);
                break;
            case 3:
                start = insertbeginning(start);
                break;
            case 4:
                start = insertend(start);
                break;
            case 5:
                start = insertmiddle(start);
                break;
            case 6:
                start = deletebeginning(start);
                break;
            case 7:
                start = deleteend(start);
                break;
            case 8:
                start = deletemiddle(start);
                break;
            }
        } while (choice != 9);
        return 0;
}
struct node *create(struct node *start)
{
    struct node *new_node, *ptr;
    int num;
    printf("\n Enter -1 to end");
    printf("\n Enter the data : ");
    scanf("%d", &num);
    while (num != -1)
    {
        new_node = (struct node *)malloc(sizeof(struct node));
        new_node->data = num;
        if (start == NULL)
        {
            new_node->next = NULL;
            start = new_node;
        }
        else
        {
            ptr = start;
            while (ptr->next != NULL)
                ptr = ptr->next;
            ptr->next = new_node;
            new_node->next = NULL;
        }
        printf("\n Enter the data : ");
        scanf("%d", &num);
    }
    return start;
}
struct node *display(struct node *start)
{
    struct node *ptr;
    ptr = start;
    while (ptr != NULL)
    {
        printf("\t %d", ptr->data);
        ptr = ptr->next;
    }
    return start;
}
struct node *insertbeginning(struct node *start)
{
    struct node *new_node;
    int num;
    printf("\n Enter the data : ");
    scanf("%d", &num);
    new_node = (struct node *)malloc(sizeof(struct node));
    new_node->data = num;
    new_node->next = start;
    start = new_node;
    return start;
}
struct node *insertend(struct node *start)
{
    struct node *ptr, *new_node;
    int num;
    printf("\n Enter the data : ");
    scanf("%d", &num);
    new_node = (struct node *)malloc(sizeof(struct node));
    new_node->data = num;
    new_node->next = NULL;
    ptr = start;
    while (ptr->next != NULL)
        ptr = ptr->next;
    ptr->next = new_node;
    return start;
}
struct node *insertmiddle(struct node *start)
{
    struct node *new_node, *ptr, *preptr;
    int num, val;
    printf("\n Enter the data : ");
    scanf("%d", &num);
    printf("\n Enter the value after which the data has to be inserted 
: ");
    scanf("%d", &val);
    new_node = (struct node *)malloc(sizeof(struct node));
    new_node->data = num;
    ptr = start;
    preptr = ptr;
    while (preptr->data != val)
    {
        preptr = ptr;
        ptr = ptr->next;
    }
    preptr->next = new_node;
    new_node->next = ptr;
    return start;
}
struct node *deletebeginning(struct node *start)
{
    struct node *ptr;
    ptr = start;
    start = start->next;
    free(ptr);
    return start;
}
struct node *deleteend(struct node *start)
{
    struct node *ptr, *preptr;
    ptr = start;
    while (ptr->next != NULL)
    {
        preptr = ptr;
        ptr = ptr->next;
    }
    preptr->next = NULL;
    free(ptr);
    return start;
}
struct node *deletemiddle(struct node *start)
{
    struct node *ptr, *preptr;
    int val;
    printf("\n Enter the value of the node which has to be deleted : 
");
    scanf("%d", &val);
    ptr = start;
    if (ptr->data == val)
    {
        start = deletebeginning(start);
        return start;
    }
    else
    {
        while (ptr->data != val)
        {
            preptr = ptr;
            ptr = ptr->next;
        }
        preptr->next = ptr->next;
        free(ptr);
        return start;
    }
}
```

OUTPUT:



```
Microsoft Windows [Version 10.0.19044.1889]
(c) Microsoft Corporation. All rights reserved.

C:\Users\shweta\Documents\Shashwat\Notepad++\DSA>gcc -o DSAexp5 DSAexp5.c

C:\Users\shweta\Documents\Shashwat\Notepad++\DSA>DSAexp5
D10A_60_Shashwat Tripathi
#############################################
Your choices are:
 1: Create list
 2: Display list
 3: Add a node at the start
 4: Add a node at the end
 5: Add a node in the middle
 6: Delete a node from the beginning
 7: Delete a node from the end
 8: Delete a node after a given node
 9: EXIT
#############################################

 Enter your choice : 1

 Enter -1 to end
 Enter the data : 23

 Enter the data : 87

 Enter the data : 69

 Enter the data : 45

 Enter the data : -1

 LINKED LIST CREATED
 Enter your choice : 3

 Enter the data : 11
```



```

 Enter the data : 11

 Enter your choice : 2
         11      23      87      69      45
 Enter your choice : 8

 Enter the value of the node which has to be deleted : 87

 Enter your choice : 4

 Enter the data : 99

 Enter your choice : 2
         11      23      69      45      99
 Enter your choice : 7

 Enter your choice : 6

 Enter your choice : 2
         23      69      45
 Enter your choice : 9

C:\Users\shweta\Documents\Shashwat\Notepad++\DSA>
```