Start here | Lab1stack.c | Lab3aLinearqueue.c | Lab3bCircularqueue.c | Lab4SLL.c | Lab5SLL.c | *Lab7DLL.c | *Lab6concate.c

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

struct Node{
    int data;
    struct Node* next;
};

struct Node* createNode(int data){
    struct Node *newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->next = NULL;
    return newNode;
}
struct Node* createList(){

    struct Node *head = NULL, *temp, *newNode;
    int n, data;

    printf("Enter number of nodes: ");
    scanf("%d",&n);

    for(int i =0;i < n; i++){
        printf("Enter data: ");
        scanf("%d",&data);
        newNode = createNode(data);
        if(head == NULL) head = newNode;
        else {
            temp = head;
            while(temp->next != NULL)
                temp = temp->next;
            temp->next = newNode;
        }
    }
    return head;
}
void display(struct Node *head){
    while(head != NULL){
        printf("%d->",head->data);
        head = head->next;
    }
    printf("NULL\n");
}
```

Start here | Lab1stack.c | Lab3aLinearqueue.c | Lab3bCircularqueue.c | Lab4SLL.c | Lab5SLL.c | *Lab7DLL.c | *Lab6concate.c

```
43  }
44  void sort(struct Node* head){
45      struct Node *i, *j;
46      int temp;
47      for(i = head; i != NULL; i = i->next){
48          for(j = i->next; j != NULL; j = j->next){
49              if(i->data > j->data){
50                  temp = i->data;
51                  i->data = j->data;
52                  j->data = temp;
53              }
54          }
55      }
56  }
57
58  struct Node *reverse(struct Node* head){
59      struct Node *prev = NULL, *curr = head,
60      *next = NULL;
61      while(curr != NULL){
62          next = curr->next;
63          curr->next  = prev;
64          prev = curr;
65          curr= next;
66      }
67          return prev;
68  }
69  struct Node *concatenate(struct Node *head1, struct Node *head2){
70      if(head1 == NULL) return head2;
71      if(head2 == NULL) return head1;
72      struct Node *temp = head1;
73      while(temp->next != NULL)
74         temp = temp->next;
75      temp->next = head2;
76      return head1;
77  }
78  int main(){
79      struct Node *list1 = NULL, *list2 = NULL;
80      printf("Create List 1: \n");
81      list1 = createList();
82
83      printf("\nCreate List 2: \n");
84      list2 = createList();
85
```

Start here | Lab1stack.c | Lab3aLinearqueue.c | Lab3bCircularqueue.c | Lab4SLL.c | Lab5SLL.c | *Lab7DLL.c | *Lab6concate.c

```c
            curr= next;
        }
            return prev;
    }

struct Node *concatenate(struct Node *head1, struct Node *head2){
    if(head1 == NULL) return head2;
    if(head2 == NULL) return head1;
    struct Node *temp = head1;
    while(temp->next != NULL)
       temp = temp->next;
    temp->next = head2;
    return head1;
}

int main(){
    struct Node *list1 = NULL, *list2 = NULL;
    printf("Create List 1: \n");
    list1 = createList();

    printf("\nCreate List 2: \n");
    list2 = createList();

    printf("\nList 1: ");
    display(list1);

    printf("List 2: ");
    display(list2);

    sort(list1);
    printf("\nList 1 After sorting: ");
    display(list1);

    list1 = reverse(list1);
    printf("List1, After Reversing: ");
    display(list1);

    list1 = concatenate(list1, list2);
    printf("After concatenation (list1 + list2): ");
    display(list1);
    return 0;
}
```

D:\Coding\C\LAB\Lab6concat

```
Create List 1:
Enter number of nodes: 4
Enter data: 23
Enter data: 2
Enter data: 65
Enter data: 12

Create List 2:
Enter number of nodes: 3
Enter data: 86
Enter data: 59
Enter data: 36

List 1: 23->2->65->12->NULL
List 2: 86->59->36->NULL

List 1 After sorting: 2->12->23->65->NULL
List1, After Reversing: 65->23->12->2->NULL
After concatenation (list1 + list2): 65->23->12->2->86->59->36->NULL

Process returned 0 (0x0)   execution time : 30.781 s
Press any key to continue.
```

Start here | Lab1stack.c | Lab3aLinearqueue.c | Lab3bCircularqueue.c | Lab4SLL.c | Lab5SLL.c | *Lab7DLL.c | *Lab6concate.c | Lab6stack.c | Lab6queue.c

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

struct node
{
    int data;
    struct node *next;
} ;
struct node *front = NULL;
struct node *rear = NULL;

void insert(struct node *ptr, int item){
    ptr = (struct node *) malloc(sizeof(struct node));
    if(ptr == NULL){
        printf("Overflow\n");
        return ;
    }
    else {
        ptr->data = item;
        if(front == NULL){
            front = ptr;
            rear = ptr;
            front->next = NULL;
            rear->next = NULL;
        }else {
            rear->next = ptr;
            rear = ptr;
            rear->next = NULL;
        }
    }
}

void deleteNode(struct node*ptr){
    if(front == NULL){
        rear = NULL;
        printf("Underflow\n");
        return ;
    }
    else {
        ptr = front;
        front = front->next;
        printf("%d is deleted\n",ptr->data);
        free(ptr);
```

Counting Loop

Lab1stack.c | Lab3aLinearqueue.c | Lab3bCircularqueue.c | Lab4SLL.c | Lab5SLL.c | *Lab7DLL.c | *Lab6concate.c | Lab6stack.c | Lab6queue.c

```
43          free(ptr);
44      }
45  }
46
47  void display(){
48
49      struct node *ptr;
50      ptr = front;
51
52      if(front == NULL && rear == NULL)
53          printf("Queue is empty\n");
54      else{
55          while(ptr->next != NULL){
56              printf("%d->",ptr->data);
57              ptr = ptr->next;
58          }
59          printf("%d\n",ptr->data);
60      }
61  }
62
63  int main(){
64      struct node * head = NULL;
65      int choice, value;
66      printf("\n1.Insert\n2.Delete\n3.Display\n4.Exit\n");
67      while(1){
68          printf("Enter your choice: ");
69          scanf("%d",&choice);
70          switch(choice){
71              case 1:printf("Enter the value to be inserted: ");
72                     scanf("%d",&value);
73                     insert(head,value);
74                     break;
75              case 2:deleteNode(head);
76                     break;
77              case 3:display();
78                     break;
79              case 4:exit(0);
80              default:printf("Invalid Choice!");
81          }
82      }
83      return 0;
84  }
85
```

D:\Coding\C\LAB\Lab6queue.

```
1.Insert
2.Delete
3.Display
4.Exit
Enter your choice: 1
Enter the value to be inserted: 23
Enter your choice: 1
Enter the value to be inserted: 46
Enter your choice: 1
Enter the value to be inserted: 69
Enter your choice: 1
Enter the value to be inserted: 92
Enter your choice: 3
23->46->69->92
Enter your choice: 2
23 is deleted
Enter your choice: 2
46 is deleted
Enter your choice: 3
69->92
Enter your choice: 1
Enter the value to be inserted: 47
Enter your choice: 1
Enter the value to be inserted: 68
Enter your choice: 4

Process returned 0 (0x0)   execution time : 45.726 s
Press any key to continue.
```

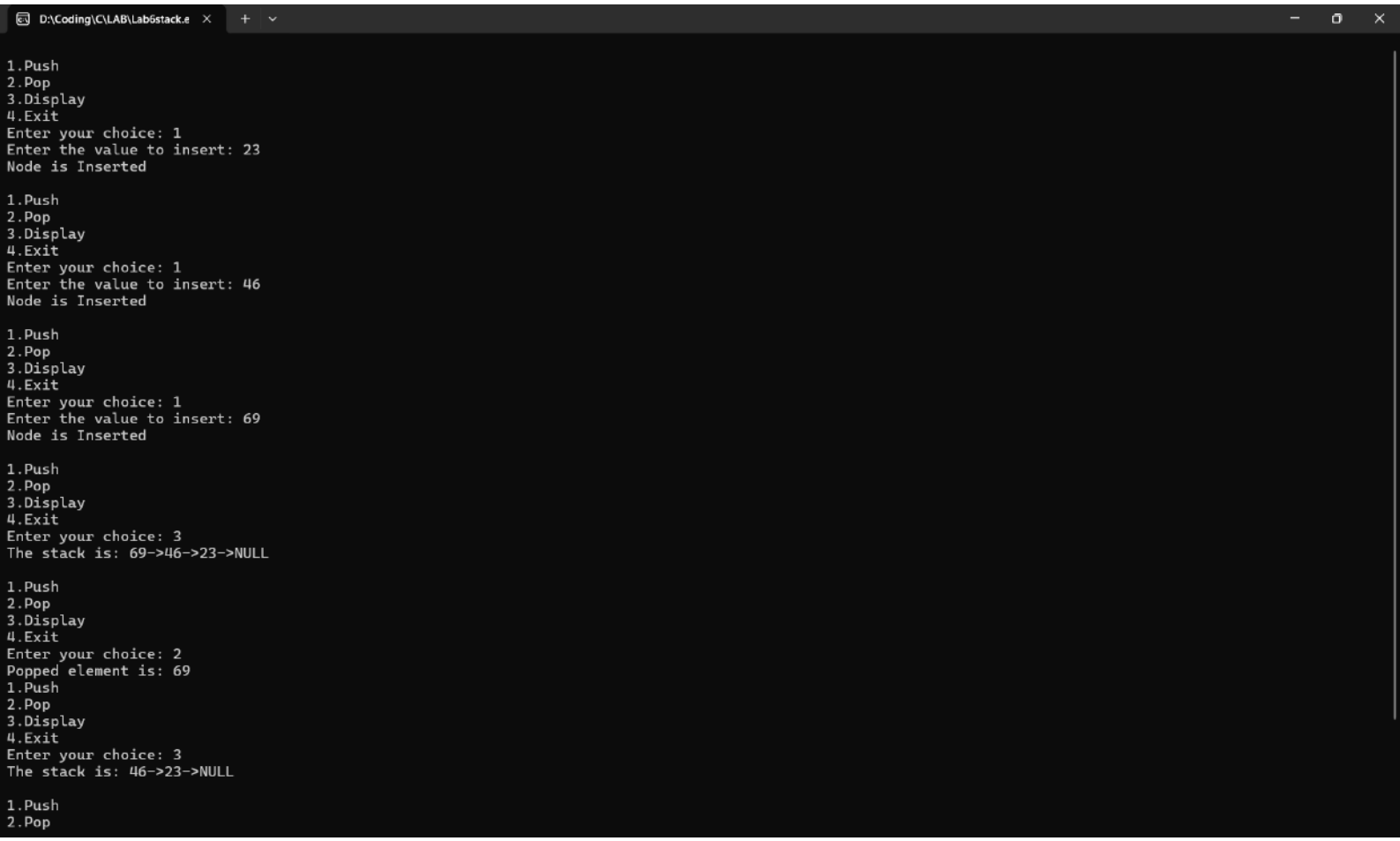
```c
#include <stdio.h>
#include <stdlib.h>

struct node
{
    int info;
    struct node *ptr;
} *top, *top1, *temp;

int count = 0;
void push(int data)
{
    if (top == NULL)
    {
        top = (struct node *)malloc(1 * sizeof(struct node));
        top->ptr = NULL;
        top->info = data;
    }
    else
    {
        temp = (struct node *)malloc(1 * sizeof(struct node));
        temp->ptr = top;
        temp->info = data;
        top = temp;
    }
    count++;
    printf("Node is Inserted\n");
}
int pop()
{
    top1 = top;
    if (top1 == NULL)
    {
        printf("Stack Underflow\n");
        return -1;
    }
    else
        top1 = top1->ptr;
    int popped = top->info;
    free(top);
    top = top1;
    count--;
    return popped;
```

Start here | Lab1stack.c | Lab3aLinearqueue.c | Lab3bCircularqueue.c | Lab4SLL.c | Lab5SLL.c | *Lab7DLL.c | *Lab6concate.c | *Lab6stack.c

```
44  }
45  void display()
46  {
47      top1 = top;
48      if (top1 == NULL)
49      {
50          printf("Stack Underflow\n");
51          return;
52      }
53      printf("The stack is: ");
54      while (top1 != NULL)
55      {
56          printf("%d-->", top1->info);
57          top1 = top1->ptr;
58      }
59      printf("NULL\n");
60  }
61
62  int main()
63  {
64      int choice, value;
65      while (1)
66      {
67          printf("\n1.Push\n2.Pop\n3.Display\n4.Exit\n");
68          printf("Enter your choice: ");
69          scanf("%d", &choice);
70          switch (choice)
71          {
72          case 1:printf("Enter the value to insert: ");
73                 scanf("%d", &value);
74                 push(value);
75                 break;
76          case 2:printf("Popped element is: %d", pop());
77                 break;
78          case 3:display();
79                 break;
80          case 4:exit(0);
81                 break;
82          default:printf("Wrong choice");
83          }
84      }
85  }
86
```


D:\Coding\C\LAB\Lab6stack.e
1.Push
2.Pop
3.Display
4.Exit
Enter your choice: 1
Enter the value to insert: 23
Node is Inserted
1.Push
2.Pop
3.Display
4.Exit
Enter your choice: 1
Enter the value to insert: 46
Node is Inserted
1.Push
2.Pop
3.Display
4.Exit
Enter your choice: 1
Enter the value to insert: 69
Node is Inserted
1.Push
2.Pop
3.Display
4.Exit
Enter your choice: 3
The stack is: 69->46->23->NULL
1.Push
2.Pop
3.Display
4.Exit
Enter your choice: 2
Popped element is: 69
1.Push
2.Pop
3.Display
4.Exit
Enter your choice: 3
The stack is: 46->23->NULL
1.Push
2.Pop