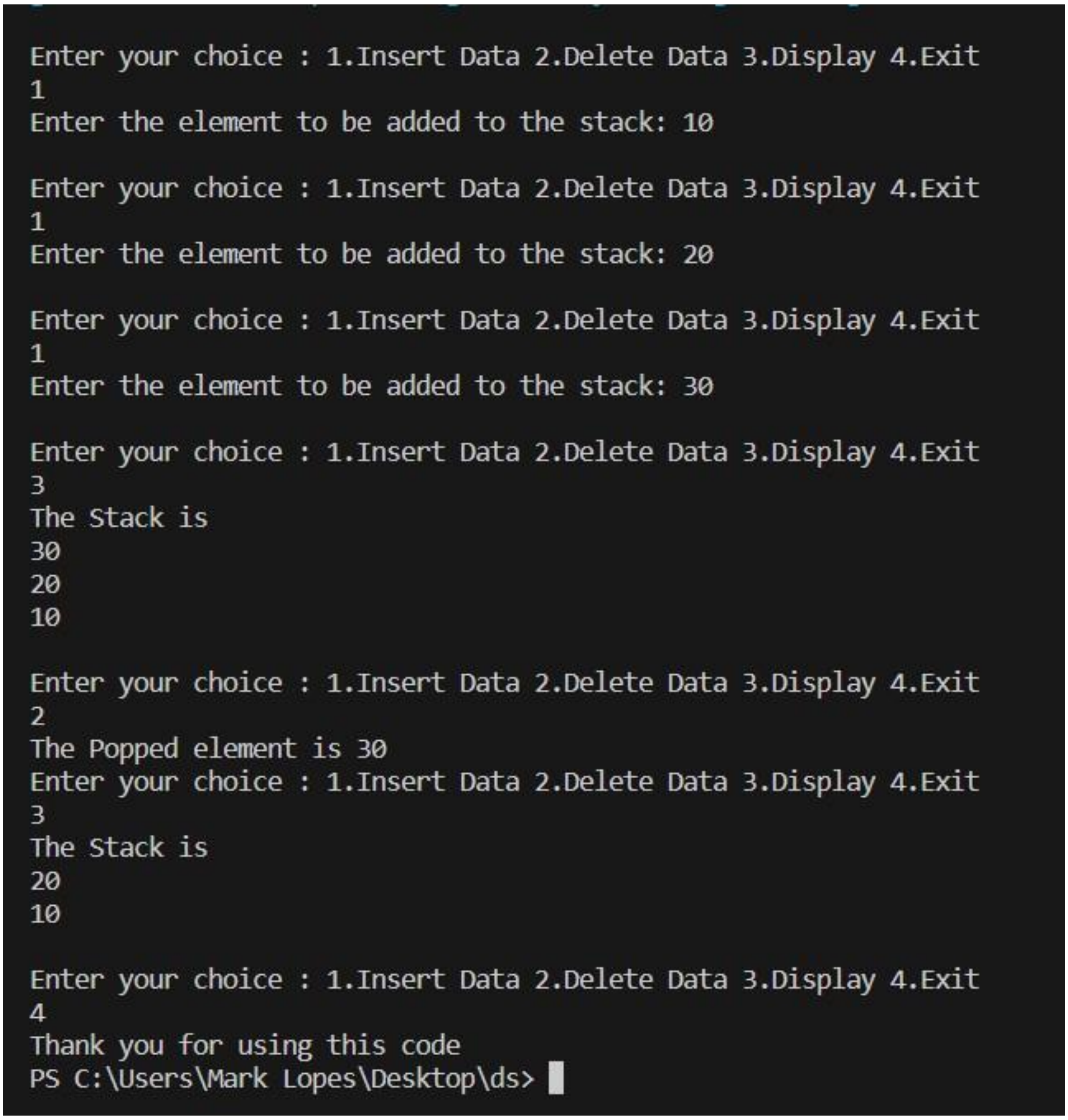
```c
// stack using LL
#include <stdio.h>
#include <stdlib.h>
typedef struct node
{
    int info;
    struct node *next;
} Node;

typedef struct
{
    Node *tos;
} StackLL;

void push(StackLL *ptr, int x)
{
    Node *p;
    p = (Node *)malloc(sizeof(Node));
    p->info = x;

    if (ptr->tos == NULL)
    {
        ptr->tos = p; // tos = p
        p->next = NULL;
    }
    else
    {
        p->next = ptr->tos; // p->next = tos
        ptr->tos = p;       // tos = p, newly created node becomes top of the
stack
    }
}

int pop(StackLL *ptr)
{
    Node *p;
    int x;

    if (ptr->tos == NULL)
    {
        printf("Stack underflow\n");
        return -1;
    }
    else
    {
        p = ptr->tos;       // p is pointer pointing to top of the stack
        x = p->info;        // store its data to
        ptr->tos = p->next; // tos = tos->next
        free(p);             // release the memory pointed by p
        return x;
    }
}

void display(StackLL s)
{
    Node *p;
    p = s.tos;
    printf("The Stack is \n");
    while (p != NULL)
    {
        printf("%d\n", p->info);
        p = p->next;
    }
}

int main()
{
    int choice, ele;
    StackLL s1;     // stack s1 created using LL
    s1.tos = NULL; // IniƟally no node

    do
    {
        printf("\nEnter your choice : 1.Insert Data 2.Delete Data 3.Display
4.Exit\n");
        scanf("%d", &choice);

        switch (choice)
        {
        case 1:
            printf("Enter the element to be added to the stack: ");
            scanf("%d", &ele);
            push(&s1, ele);
            break;

        case 2:
            printf("The Popped element is %d", pop(&s1));
            break;

        case 3:

            display(s1);
            break;

        case 4:
            printf("Thank you for using this code\n");
        }
    } while (choice != 4);
    return 0;
}
```



```
Enter your choice : 1.Insert Data 2.Delete Data 3.Display 4.Exit
1
Enter the element to be added to the stack: 10

Enter your choice : 1.Insert Data 2.Delete Data 3.Display 4.Exit
1
Enter the element to be added to the stack: 20

Enter your choice : 1.Insert Data 2.Delete Data 3.Display 4.Exit
1
Enter the element to be added to the stack: 30

Enter your choice : 1.Insert Data 2.Delete Data 3.Display 4.Exit
3
The Stack is
30
20
10

Enter your choice : 1.Insert Data 2.Delete Data 3.Display 4.Exit
2
The Popped element is 30
Enter your choice : 1.Insert Data 2.Delete Data 3.Display 4.Exit
3
The Stack is
20
10

Enter your choice : 1.Insert Data 2.Delete Data 3.Display 4.Exit
4
Thank you for using this code
PS C:\Users\Mark Lopes\Desktop\ds>
```