Aaryan Sharma – 16010123012

Reversing a stack using Linked Lists

Code –

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

struct Node {
   int value;
   struct Node* nextNode;
};

struct Node* topNode = NULL;

void addNode(int number) {
   struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
   newNode -> value = number;
   newNode -> nextNode = topNode;
   topNode = newNode;
}

void reverseStack() {
   struct Node* previousNode = NULL;
   struct Node* currentNode = topNode;
   struct Node* nextNode = NULL;

   while (currentNode != NULL) {
      nextNode = currentNode->nextNode;
      currentNode->nextNode = previousNode;
      previousNode = currentNode;
```

```
        currentNode = nextNode;
    }
    topNode = previousNode;
}

void displayStack() {
    struct Node* tempNode = topNode;
    while (tempNode != NULL) {
        printf("%d ", tempNode->value);
        tempNode = tempNode->nextNode;
    }
}

int main() {
    int numElements, element;
    printf("Enter the number of elements: ");
    scanf("%d", &numElements);
    for (int i = 0; i < numElements; i++) {
        printf("Enter %d element : ", i + 1);
        scanf("%d", &element);
        addNode(element);
    }
    printf("\nOriginal Linked List: ");
    displayStack();
    reverseStack();
    printf("\nReversed Linked List: ");
    displayStack();
    return 0;
}
```

Output –

```
Enter the number of elements: 5
Enter 1 element : 1
Enter 2 element : 2
Enter 3 element : 3
Enter 4 element : 4
Enter 5 element : 5

Original Linked List: 5 4 3 2 1
Reversed Linked List: 1 2 3 4 5
```

```
Enter the number of elements: 6
Enter 1 element : 123
Enter 2 element : 45
Enter 3 element : 23
Enter 4 element : 65
Enter 5 element : 2
Enter 6 element : 54

Original Linked List: 54 2 65 23 45 123
Reversed Linked List: 123 45 23 65 2 54
```

```
Enter the number of elements: 3
Enter 1 element : 457
Enter 2 element : 7834
Enter 3 element : 585

Original Linked List: 585 7834 457
Reversed Linked List: 457 7834 585
```