S.No: 4	Exp. Name: ***Stack Implementation Using Linked Lists***	Date:2025-08-17

Aim:

Write a program to implement a stack using linked lists.

Input Format:

The user is presented with a menu of options and provides input according to the desired operation:

1. **Push Operation:**
   Input: Integer value to be pushed onto the stack.
2. **Pop Operation:**
   No additional input is required.
3. **Display Operation:**
   No additional input is required.
4. **Is Empty Operation:**
   No additional input is required.
5. **Peek Operation:**
   No additional input is required.
6. **Exit Operation:**
7. No additional input is required.

Output Format:

The output will vary depending on the selected option:

1. **Push Operation:**
   If the stack is not full (no overflow), the output will be: "Successfully pushed."
2. **Pop Operation:**
   If the stack is not empty, it will print: "Popped value = X" where X is the value removed from the stack. If the stack is empty, it will print: "Stack is underflow."
3. **Display Operation:**
   If the stack is not empty, it will print: "Elements of the stack are : X Y Z ..." where X, Y, Z, etc., are the elements from top to bottom. If the stack is empty, it will print: "Stack is empty."
4. **Is Empty Operation:**
   If the stack is empty, it will print: "Stack is empty." If the stack is not empty, it will print: "Stack is not empty."
5. **Peek Operation:**
   If the stack is not empty, it will print: "Peek value = X" where X is the top element of the stack.If the stack is empty, it will print: "Stack is underflow."
6. **Exit Operation:**
7. The program terminates with no additional output.

Note:

- The partial code has been provided to you in the editor, you are required to fill in the missing code.

Source Code:

StackUsingLL.c

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

int main() {
   int op, x;
   while(1) {
```

```
      printf("1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit\n");
      printf("Enter your option : ");
      scanf("%d", &op);
      switch(op) {
         case 1:
            printf("Enter element : ");
            scanf("%d", &x);
            push(x);
            break;
         case 2:
            pop();
            break;
         case 3:
            display();
            break;
         case 4:
            isEmpty();
            break;
         case 5:
            peek();
            break;
         case 6:
            exit(0);
      }
   }
}
```

StackOperationsLL.c

```
struct stack {
   int data;
   struct stack *next;
};

typedef struct stack *stk;
stk top = NULL;

stk push(int x) {
   stk newNode = (stk )malloc(sizeof(stk));
   if(!newNode){
      printf("Stack overflow\n");
      return NULL;
   }
   newNode->data = x;
   newNode->next = top;
   top = newNode;
   printf("Successfully pushed.\n");
   return newNode;

}

stk pop() {
   if(!top){
      printf("Stack is underflow.\n");
```

```
      return NULL;
   }
   stk temp = top;
   int data = temp->data;
   top = top->next;
   printf("Popped value = %d\n", data);
   return temp;

}

void peek() {
   if(!top){
      printf("Stack is underflow.\n");
      return;
   }
   printf("Peek value = %d\n", top->data);

}

void isEmpty() {
   if(!top)
      printf("Stack is empty.\n");
   else
      printf("Stack is not empty.\n");

}

void display() {
   stk temp = top;
   if(temp == NULL) {
      printf("Stack is empty.\n");
   } else {
      printf("Elements of the stack are : ");
      while(temp != NULL) {
         printf("%d ", temp -> data);
         temp = temp -> next;
      }
      printf("\n");
   }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 1
Enter your option : 1
Enter element : 33
Successfully pushed. 1
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 1
Enter your option : 1
Enter element : 22
Successfully pushed. 1

1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 1
Enter your option : 1
Enter element : 55
Successfully pushed. 1
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 1
Enter your option : 1
Enter element : 66
Successfully pushed. 3
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 3
Enter your option : 3
Elements of the stack are : 66 55 22 33 2
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 2
Enter your option : 2
Popped value = 66 2
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 2
Enter your option : 2
Popped value = 55 3
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 3
Enter your option : 3
Elements of the stack are : 22 33 5
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 5
Enter your option : 5
Peek value = 22 4
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 4
Enter your option : 4
Stack is not empty. 6
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 6
Enter your option : 6

Test Case - 2
User Output
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 2
Enter your option : 2
Stack is underflow. 3
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 3
Enter your option : 3
Stack is empty. 5
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 5
Enter your option : 5
Stack is underflow. 4
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 4
Enter your option : 4
Stack is empty. 1
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 1
Enter your option : 1
Enter element : 23
Successfully pushed. 1
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 1
Enter your option : 1
Enter element : 24

```
Successfully pushed. 3
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 3
Enter your option :  3
Elements of the stack are : 24 23  5
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 5
Enter your option :  5
Peek value = 24 2
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 2
Enter your option :  2
Popped value = 24 2
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 2
Enter your option :  2
Popped value = 23 2
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 2
Enter your option :  2
Stack is underflow. 4
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 4
Enter your option :  4
Stack is empty. 6
1.Push 2.Pop 3.Display 4.Is Empty 5.Peek 6.Exit 6
Enter your option :  6
```