IMPLEMENTATION OF STACK USING ARRAY:

```
//STACK USING ARRAY
#include <stdio.h>
#include <stdlib.h>

#define MAX 100

int stack[MAX];
int top = -1;

void push(int data)
{
    if (top == MAX - 1)
    {
        printf("Stack overflow. Unable to push %d onto stack.\n", data);
    }
    else
    {
        stack[++top] = data;
        printf("Pushed %d onto stack.\n", data);
    }
}

int pop()
{
    if (top == -1)
    {
        printf("Stack underflow. Unable to pop from stack.\n");
        return -1;
    }
    else
    {
        int data = stack[top--];
        printf("Popped %d from stack.\n", data);
        return data;
    }
}

int peek()
{
    if (top == -1)
    {
        printf("Stack is empty. No element to peek.\n");
        return -1;
    }
    else
    {
        return stack[top];
    }
}

int isEmpty()
{
    return top == -1;
}

void display()
{
    if (top == -1)
    {
        printf("Stack is empty.\n");
    }
    else
    {
        printf("Stack elements: ");
        for (int i = top; i >= 0; i--)
        {
            printf("%d ", stack[i]);
        }
        printf("\n");
    }
}

void main()
{
    int choice, data;
    while (1)
    {
        printf("\nStack Operations Menu:\n");
        printf("1. Push\n");
        printf("2. Pop\n");
        printf("3. Peek\n");
        printf("4. Display\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch (choice)
        {
            case 1:
                printf("Enter data to push: ");
                scanf("%d", &data);
                push(data);
                break;
            case 2:
                pop();
                break;
            case 3:
                data = peek();
                if (data != -1) {
                    printf("Top element is %d\n", data);
                }
                break;
            case 4:
                display();
                break;
            case 5:
                exit(0);
            default:
                printf("Invalid choice. Please try again.\n");
        }
    }
}
```

IMPLEMENTATION OF STACK USING LINKED LIST:

```
//STACK USING LINKED LIST
#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;
}
```

```
int isEmpty(struct Node* top)
{
   return top == NULL;
}

void push(struct Node** top, int data)
{
   struct Node* newNode = createNode(data);
   newNode->next = *top;
   *top = newNode;
   printf("Pushed %d onto stack.\n", data);
}

int pop(struct Node** top)
{
   if (isEmpty(*top))
   {
      printf("Stack underflow. Unable to pop from stack.\n");
      return -1;
   }
   else
   {
      struct Node* temp = *top;
      int poppedData = temp->data;
      *top = (*top)->next;
      free(temp);
      printf("Popped %d from stack.\n", poppedData);
      return poppedData;
   }
}

int peek(struct Node* top)
{
   if (isEmpty(top))
   {
      printf("Stack is empty. No element to peek.\n");
      return -1;
   }
   else
   {
      return top->data;
   }
}

void display(struct Node* top)
{
   if (isEmpty(top))
   {
        printf("Stack is empty.\n");
    }
    else
    {
        struct Node* temp = top;
        printf("Stack elements: ");
        while (temp != NULL)
        {
            printf("%d ", temp->data);
            temp = temp->next;
        }
        printf("\n");
    }
}

void main()
{
    struct Node* top = NULL;
    int choice, data;
    while (1)
    {
        printf("\nStack Operations Menu:\n");
        printf("1. Push\n");
        printf("2. Pop\n");
        printf("3. Peek\n");
        printf("4. Display\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch (choice)
        {
            case 1:
                printf("Enter data to push: ");
                scanf("%d", &data);
                push(&top, data);
                break;
            case 2:
                pop(&top);
                break;
            case 3:
                data = peek(top);
                if (data != -1) {
                    printf("Top element is %d\n", data);
                }
                break;
            case 4:
                display(top);
                break;
            case 5:
                exit(0);
            default:
                printf("Invalid choice. Please try again.\n");
        }
    }
}
```