WAP to Implement Single Link List to simulate Stack

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

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

struct Node *top = NULL;

void push(int value)
{
    struct Node *newNode = (struct Node *)malloc(sizeof(struct Node));
    if (newNode == NULL)
    {
        printf("Memory allocation failed.\n");
        return;
    }
    newNode->data = value;
    newNode->next = top;
    top = newNode;
    printf("Element %d pushed to the stack.\n", value);
}

void pop()
{
    if (top == NULL)
    {
```

```
        printf("Stack underflow. Cannot pop from an empty stack.\n");
        return;
    }
    struct Node *temp = top;
    top = top->next;
    printf("Element %d popped from the stack.\n", temp->data);
    free(temp);
}

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

int main()
{
    int choice, value;

    while (1)
```

```
{
    printf("\nStack Operations:\n");
    printf("1. Push\t");
    printf("2. Pop\t");
    printf("3. Display\t");
    printf("4. Exit\n");

    printf("Enter your choice: ");
    scanf("%d", &choice);

    switch (choice)
    {
    case 1:
        printf("Enter the value to push: ");
        scanf("%d", &value);
        push(value);
        break;
    case 2:
        pop();
        break;
    case 3:
        display();
        break;
    case 4:
        printf("Exiting the stack program.\n");
        exit(0);
    default:
        printf("Invalid choice. Please enter a valid option.\n");
    }
}
```

```
    return 0;
}
```

OUTPUT

```
Stack Operations:
1. Push 2. Pop  3. Display      4. Exit
Enter your choice: 1
Enter the value to push: 40
Element 40 pushed to the stack.

Stack Operations:
1. Push 2. Pop  3. Display      4. Exit
Enter your choice: 1
Enter the value to push: 60
Element 60 pushed to the stack.

Stack Operations:
1. Push 2. Pop  3. Display      4. Exit
Enter your choice: 1
Enter the value to push: 80
Element 80 pushed to the stack.

Stack Operations:
1. Push 2. Pop  3. Display      4. Exit
Enter your choice: 1
Enter the value to push: 100
Element 100 pushed to the stack.

Stack Operations:
1. Push 2. Pop  3. Display      4. Exit
Enter your choice: 3
Stack elements: 100
80
60
40
20


Stack Operations:
1. Push 2. Pop  3. Display      4. Exit
Enter your choice: 2
Element 100 popped from the stack.

Stack Operations:
1. Push 2. Pop  3. Display      4. Exit
Enter your choice: 2
Element 80 popped from the stack.
```

```
Stack Operations:
1. Push 2. Pop  3. Display      4. Exit
Enter your choice: 3
Stack elements: 60
40
20


Stack Operations:
1. Push 2. Pop  3. Display      4. Exit
Enter your choice: 4
Exiting the stack program.
```