Problems on Stacks

1. What Is a Stack? Explain Its Characteristics.
2. What Are the Applications of a Stack?
3. How Is a Stack Implemented?

4. Write a C program to implement a stack using an array with push and pop operations.

Expected Output:

```
Elements in the stack are: 3 5 4 3 2 1
```

5. Write a C program to implement a stack using a singly linked list.

Expected Output:

```
Push data 1
Push data 2
Push data 3
Push data 4
Pop data: 4
Pop data: 3
Pop data: 2
Pop data: 1
Check a stack is empty or not?
Stack is empty!
```

6. Write a C program to check a stack is full or not using an array with push and pop operations.

Expected Output:

```
Stack size: 3
Original Stack: 1 2 3
Push another value and check if the stack is full or not!
Stack is full!
Stack size: 3
Original Stack: 10 20
Check the said stack is full or not!
Stack is not full!
```

7. Write a C program that accepts a string and reverse it using a stack.

Expected Output:

```
Input a string: w3resource
Reversed string using a stack is: ecruoser3w
```

8. Write a C program to implement two stacks in a single array and performs push and pop operations for both stacks.

Expected Output:

```
 Elements in Stack-1 are: 50 40 30 10
Elements in Stack-2 are: 70 60 50 40 20
```

9. Write a C program that checks whether a string of parentheses is balanced or not using stack.

Expected Output:

```
Input an expression in parentheses: {[])
The expression is not balanced.
 ----------------------------------------
 Input an expression in parentheses: ((()))
The expression is balanced.
 ----------------------------------------
 Input an expression in parentheses: ())
The expression is not balanced.
 ----------------------------------------
 Input an text of parentheses: ([]){}[[(){}]{}]
The expression is balanced.
 ----------------------------------------
 Input an expression in parentheses: [(]))
The expression is not balanced.
```