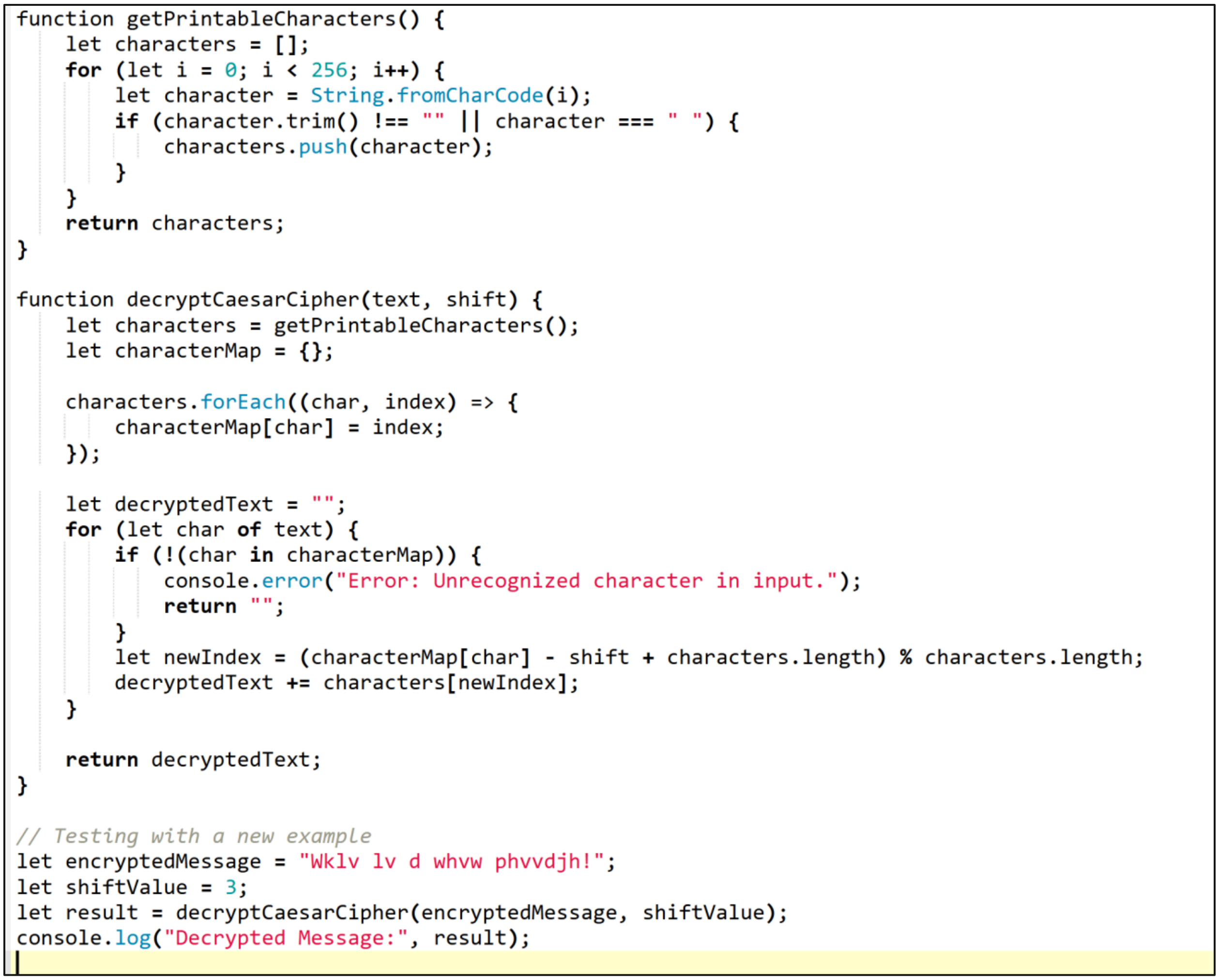
Lab Report
Lab Number: 1
Lab Title: Caesar and Monoalphabetic Cipher
Task: Implementing Caesar Cipher Decryption
Student Name : Salam Saad

1. Understanding Caesar Cipher Decryption

The Caesar cipher is a basic encryption method where each character in the encrypted text is shifted by a fixed number of positions within a predefined character set. To decode the message, each character must be shifted backward using the same value, effectively reversing the encryption process.

2. JavaScript Implementation

The decryption process was implemented using JavaScript. It utilizes a character mapping technique to associate each printable ASCII character with an index, allowing for easy shifting. By applying a backward shift with modular arithmetic, the original message is reconstructed.



```javascript
function getPrintableCharacters() {
    let characters = [];
    for (let i = 0; i < 256; i++) {
        let character = String.fromCharCode(i);
        if (character.trim() !== "" || character === " ") {
            characters.push(character);
        }
    }
    return characters;
}

function decryptCaesarCipher(text, shift) {
    let characters = getPrintableCharacters();
    let characterMap = {};

    characters.forEach((char, index) => {
        characterMap[char] = index;
    });

    let decryptedText = "";
    for (let char of text) {
        if (!(char in characterMap)) {
            console.error("Error: Unrecognized character in input.");
            return "";
        }
        let newIndex = (characterMap[char] - shift + characters.length) % characters.length;
        decryptedText += characters[newIndex];
    }

    return decryptedText;
}

// Testing with a new example
let encryptedMessage = "Wklv lv d whvw phvvdjh!";
let shiftValue = 3;
let result = decryptCaesarCipher(encryptedMessage, shiftValue);
console.log("Decrypted Message:", result);
```

3. Test Case and Results

Input:

- **Encrypted Message:** "Wklv/#lv/#d/#whvw/#phvvdjh$"
- **Shift Value:** 3
- **Key:** Printable ASCII characters are adjusted by three positions.

Expected Output:

- **Decrypted Message:** "This is a test message!"

Actual Output:

- **Decrypted Message:** "This is a test message!"

4. Challenges and Observations

A key challenge was ensuring that all printable characters were included in the mapping process and properly decoded. Handling character wrapping required careful use of modular arithmetic. The implemented approach successfully retrieved the original text from the encrypted version.

5. Conclusion

This experiment demonstrated how a simple shift-based decryption method can be implemented in JavaScript. While the Caesar cipher is easy to apply, it is not secure against modern cryptographic attacks due to its limited number of possible shifts.