## a) String Compression

Implement a method to perform string compression. E.g. 'aabcccccaaa' should be a2b1c5a3. The code to implement this is given in the link - https://www.educative.io/answers/string-compression-using-run-length-encoding

Think about memory occupied and how it can be improved.

Bonus 1:

The answer should be taken into second compressor and compress further.

E.g. a2b2c1a3c3 should become ab2clac3

Bonus 2: decompress2

ab2c1ac3 should return aabbcaaaccc.

Think about how you will test this code.

## Solution:

```
def compress_string(string):
    compressed = []  # Stores the compressed representation
    count = 1  # Initialize the count of consecutive characters to 1
    for i in range(1, len(string)):
        if string[i] == string[i - 1]:
            count += 1  # Increment the count if the current character is the same as the previous character
        else:
            compressed.append(string[i - 1])  # Append the previous character
            compressed.append(str(count))  # Append its count
            count = 1  # Reset the count to 1 for the new character
 # Add the last character and its count to the compressed list
    compressed.append(string[-1])
    compressed.append(str(count))
    compressed_string = ''.join(compressed)  # Join the compressed list into a string
    # Check if the compressed string is shorter than the original string
    if len(compressed_string) < len(string):
        return compressed_string  # Return the compressed string if it is shorter
    else:
```

```
        return string  # Return the original string if the compressed string is not shorter


def compress_string_twice(string):
    compressed = compress_string(string)  # Compress the string once using the compress_string function
    return compress_string(compressed)  # Compress the result again


def decompress_string_twice(string):
    decompressed = decompress_string(string)  # Decompress the string once using the decompress_string function
    return decompress_string(decompressed)  # Decompress the result again


def decompress_string(string):
    decompressed = []  # Stores the decompressed representation
    i = 0
    while i < len(string):
        char = string[i]
        i += 1
        count = ""  # Initialize an empty count string
        while i < len(string) and string[i].isdigit():
            count += string[i]  # Append digits to the count string
            i += 1
        if count:
            decompressed.append(char * int(count))  # Repeat the character by the count value
        else:
            decompressed.append(char)  # If no count value, simply append the character
    return ''.join(decompressed)  # Join the decompressed list into a string
input_string = input("Enter a string: ")
compressed_once = compress_string(input_string)  # Compress the input string once
print("Compressed once:", compressed_once)
compressed_twice = compress_string_twice(input_string)  # Compress the input string twice
print("Compressed twice:", compressed_twice)
```

```
decompressed_twice = decompress_string_twice(compressed_twice)  # Decompress the doubly compressed string

print("Decompressed twice:", decompressed_twice)
```

## Explanation:

**compress_string(string):** This function takes a string as input and compresses it by replacing consecutive repeated characters with the character followed by its count. It iterates through the string and keeps track of the count of consecutive characters. When a new character is encountered or the end of the string is reached, it appends the previous character and its count to the compressed list. Finally, it joins the compressed list into a string and checks if the compressed string is shorter than the original string. It returns the compressed string if it is shorter, otherwise, it returns the original string.

**compress_string_twice(string):** This function compresses the input string twice by calling the compress_string function twice. It first compresses the string using compress_string and then compresses the result again by calling compress_string on the compressed string. It returns the doubly compressed string.

**decompress_string(string):** This function takes a compressed string as input and decompresses it by repeating each character according to its count. It iterates through the string and extracts the character and count information. If a count is present, it appends the character repeated by the count to the decompressed list. If there is no count, it simply appends the character. Finally, it joins the decompressed list into a string and returns the decompressed string.

**decompress_string_twice(string):** This function decompresses the input string twice by calling the decompress_string function twice. It first decompresses the string using decompress_string and then decompresses the result again by calling decompress_string on the decompressed string. It returns the doubly decompressed string.

**input_string:** This variable takes input from the user, representing the string to be compressed and decompressed.

**compressed_once:** This variable stores the result of compressing the input string once by calling compress_string(input_string).

**compressed_twice:** This variable stores the result of compressing the input string twice by calling compress_string_twice(input_string).

**decompressed_twice:** This variable stores the result of decompressing the doubly compressed string by calling decompress_string_twice(compressed_twice).

The final print statements display the compressed and decompressed strings .

**For this Question I was knowing how to do the string compression but I was not knowing how to do the bonus 1 String compression and I check for it in the ChatGPT and I found the solution , I went through the code and understood it and implemented the code.**