DEFLATE CODING WITH FREQUENCY TABLE

```
input_string = 'BANANA$';
window_size = 10;

% Encode the input string
deflate_output = deflate_encode(input_string, window_size);

% Display the frequency table and compressed data
disp('Frequency Table:');
```

```
Frequency Table:
```

```
disp(deflate_output.frequency_table);
```

```
     0     2    36    65    66    78
     8     2     1     2     1     1
```

```
disp('Compressed Data:');
```

```
Compressed Data:
```

```
disp(deflate_output.compressed_data);
```

```
  Columns 1 through 20

     1     1     0     1     1     1     1     1     0     1     0     1     1     0     0     0     0     0     1

  Columns 21 through 31

     0     1     0     1     0     1     1     0     1     1     0
```

```
% Calculate compression ratio
original_size = length(input_string) * 8; % Original size in bits (8 bits per character)
compressed_size = length(deflate_output.compressed_data);
compression_ratio = original_size / compressed_size;

disp(['Original Size: ', num2str(original_size), ' bits']);
```

```
Original Size: 56 bits
```

```
disp(['Compressed Size: ', num2str(compressed_size), ' bits']);
```

```
Compressed Size: 31 bits
```

```
disp(['Compression Ratio: ', num2str(compression_ratio), ':1']);
```

```
Compression Ratio: 1.8065:1
```

```
% Decode and verify
decoded_output = deflate_decode(deflate_output, window_size);
disp(['Original: ', input_string]);
```

```
Original: BANANA$
```

```
disp(['Decoded:  ', char(decoded_output)]); % Convert numerical output back to characters
```

```
Decoded:  BANANA$
```