Introduction

This is a Python script that implements a straightforward program for encoding and decoding Morse code. It also handles errors like a key error. After encountering an error, the program automatically detects the issue and prints an appropriate message to the user.

Explanation of the Code

"When you run this code, it will display the welcome message from the print_intro() function. After that, it will prompt to choose whether you want to encode or decode using the get_input() function. If you enter 'e', it will allow you to input a plain text message to encode into Morse code. The program will then encode the message and display it in Morse code. It will then prompt you if you want to encode or decode another message. If you press 'y', the process will repeat. If you press 'n', the program will display the 'Thanks for using the program, goodbye!' message from the print_outro() function and terminate.

If you choose to decode by entering 'd', you'll be prompted to input a Morse code to decode. The program will then attempt to decode the message into plain text and display it. As with encoding, you'll be prompted to encode or decode another message. Again, selecting 'y' will allow you to continue, and 'n' will end the program.

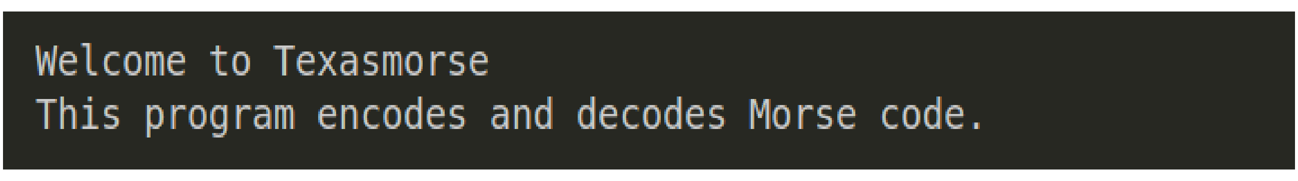
Data Structure used in this Code:
1. Tuple
2. Dictionary
3. String
4. Boolean

Function used in this Code:
1.encode():
2.decode():
3.get_input():
4.intro()
5.outro()

Structure of the Code

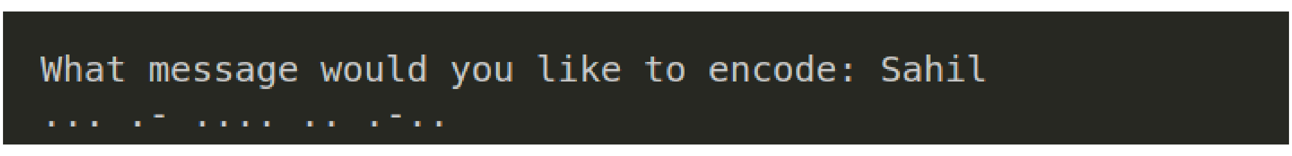
```
# Define a dictionary to map Morse code characters to their corresponding letters, numbers, and symbols
MORSE_CODE = {
    # ... (Morse code mappings)
}
```

```
1.def print_intro(): # “Function to print the introduction message”
    # ...statements
Output:
```



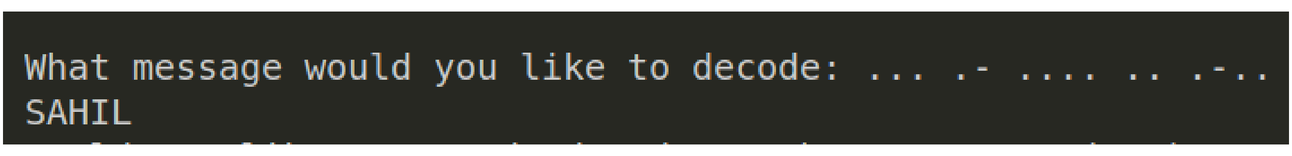
```
Welcome to Texasmorse
This program encodes and decodes Morse code.
```

```
3.def encode(message): # “Function to encode a given message into Morse code”
    # ...statements
Output:
```



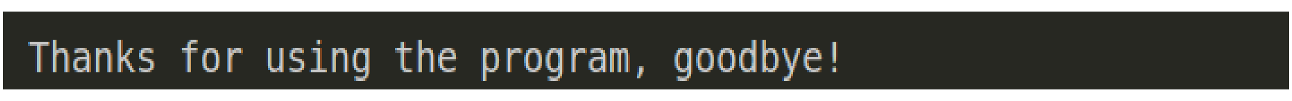
```
What message would you like to encode: Sahil
... .- .... .. .-..
```

```
3.def decode(message): # “Function to decode a given Morse code message into regular text”
    # ...statements
Output:
```



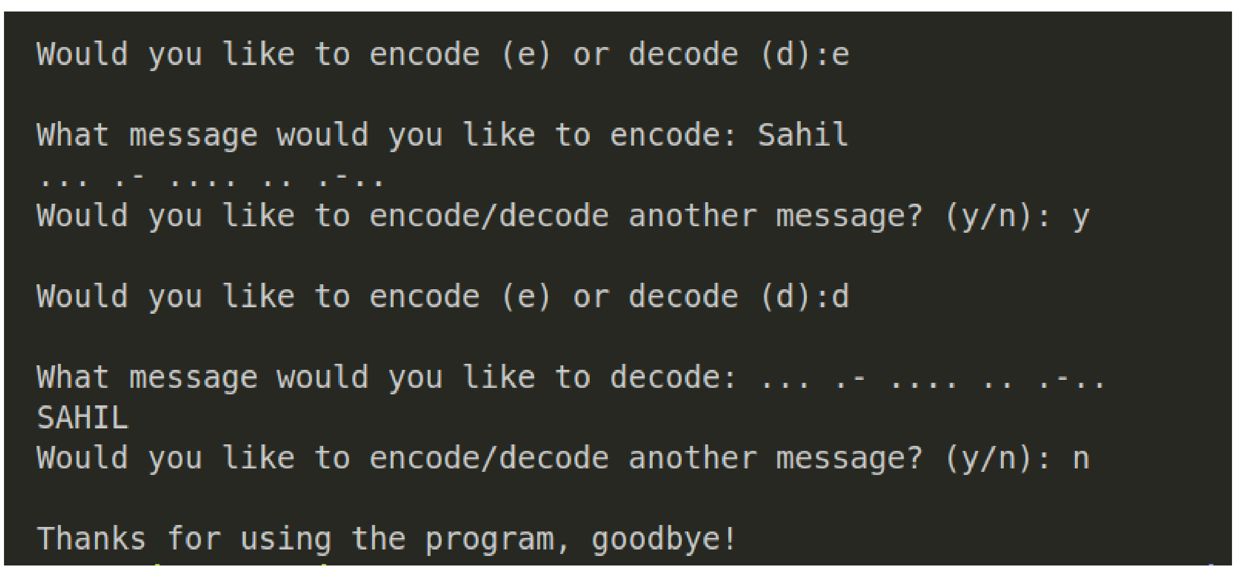
```
What message would you like to decode: ... .- .... .. .-..
SAHIL
```

```
4.def print_outro(): # “Function to print the outro message”
    # ...Statements\
Output:
```



```
Thanks for using the program, goodbye!
```

```
2.def get_input(): # “Function to input text for encoding or decoding”
    # ...Statements
Output:
```

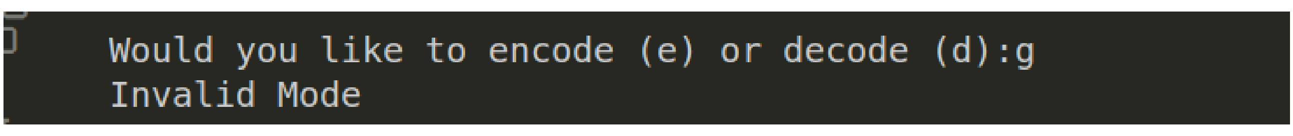


```
Would you like to encode (e) or decode (d):e

What message would you like to encode: Sahil
... .- .... .. .-..
Would you like to encode/decode another message? (y/n): y

Would you like to encode (e) or decode (d):d

What message would you like to decode: ... .- .... .. .-..
SAHIL
Would you like to encode/decode another message? (y/n): n

Thanks for using the program, goodbye!
```

```
# “Entry point of the program”
if __name__ == '__main__':
    print_intro()  # “Print the introduction”
    get_input()    # “Input text for encoding or decoding”
```

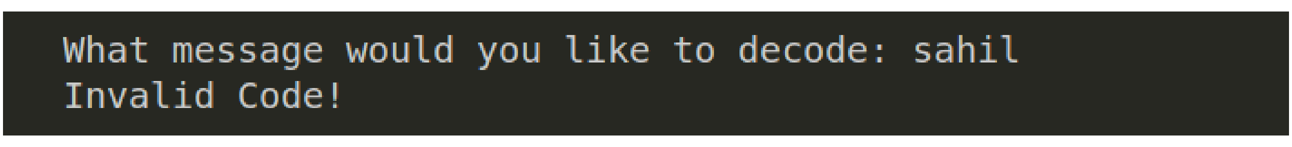
Error Handling

1. **If we enter characters other then ‘e’ or ‘d’**
Output:



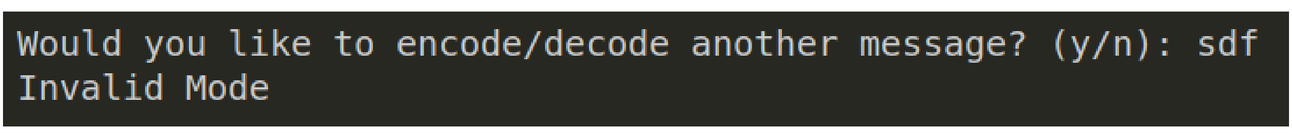
```
Would you like to encode (e) or decode (d):g
Invalid Mode
```

2. **if we enter characters to decode instated of morse code:**
Output:



```
What message would you like to decode: sahil
Invalid Code!
```

3. **if we enter characters others then ‘y’ or ‘no’:**
Output:



```
Would you like to encode/decode another message? (y/n): sdf
Invalid Mode
```

After displaying the error message to the user, another prompt will appear, allowing us to enter the correct characters. The code will then work successfully."