User Documentation for the Morse Code Decoder

Contents

1. Description of the program

The program is used to decode Morse code sequences that represent words and sentences in English. The dots, dashes and spaces in the Morse Code system are represented by 0, 1 and '*' respectively. Each character represented by the Morse Code system is separated by a single space, and each word is separated by three spaces. The program decodes the Morse Code sequences entered by the user and displays the decoded message. If there is an error in any sequence entered by the user, the program displays an error. Note that the input sequence also has other minimum requirements: it should have at least one set of three '***' and should terminate with a punctuation mark. If there was no error in the input, the program displays the decoded message and then presents a menu asking the user for the level of analysis for the decoded message. The different levels for analyses are characters, words, and sentences. Based on the level selected by the user, the program displays the occurrences of each character / word / type of sentence.

Note that the programming language used for the program is **Python 3.6**.

2. Layout of the program

The program is divided into 5 tasks to show the progression of the program. The table below summarizes the different tasks in the program:

Task Number	File name	Detail
Task 1	decoder	Building a class for Morse Code Decoder
Task 2	character_analysis	Building a class for analysis decoded characters
Task 3	word_analysis	Building a class for analysis decoded words
Task 4	sentence_analysis	Building a class for analysis decoded sentences
Task 5	main	Putting all the classes together

The first 4 files contain the classes needed for decoding the Morse code sequence and analysis of the decoded message. The fifth file is the most important file from users' point of view, this file is where the "actual program" runs.

3. Limitations of the program

- The valid punctuation marks for the program are " , " , " ? " and " . ". Any other punctuation mark or special character is considered as an invalid input for the program.
- The program does not check for proper words and sentences in English with correct spellings and grammar. This also affects the number of clauses in the decoded message returned by the program.

  However, the program currently has the following simple grammar checks:

  - A sentence cannot begin with a punctuation mark. For example, the program will return an error for the decoded message – "? How are you?"
  - A sentence cannot have two consecutive punctuation marks. For example, the program will return as error for the decoded message – "Hello, how are you?? Is everything fine?"

4. Assumptions of the program

- The program does not consider leading and trailing spaces in the input sequence. The leading and trailing spaces are removed before decoding the input sequences. For example, the program would translate the input – "***011*0000*111***01*11***00***001100***" as "WHO AM I ?" and not as " Who am I ? "
  - Note that the number of "*" between sequences needs to be correct: either 1 or 3. The program will return an error if there are two or more than 3 "*" between sequences.
- In the analysis of sentences for the decoded message, the number of clauses are computed based on the number of commas as follows:
  - If there is a comma in a sentence ending with any punctuation mark, the number of clauses will be two for the sentence. For example, the program will return 2 clauses for the sentence – "The weather is great, but I am working on the assignment."
  - If there are more than one comma in a sentence ending with any punctuation mark, the number of clauses will be the sum of the commas and the last punctuation mark (which could also be a comma). For example, the program will return 5 clauses for the sentence – "A, B, C, D, E and F are good friends."
  - If there only one comma in the sentence which is at the end of the sentence, the number of clauses will be 1. For example, the program will return 1 clause for the sentence – "The weather is great, "