Building a Simple Morse Code Decoder with Python Programming

This program is designed to take input in the combination of zeros and ones from the user which we will be called as morse code. The program is designed such that each Morse code is separated by * from next Morse code.

The program is divided into four subprograms Task1, Task2, Task3, Task4.

Task1 – A morse code dictionary is defined in Task1. This is a dictionary containing morse code as keys and characters as their values. In total there are 36 Morse codes (26 alphabets [A-Z] and 10 numbers [0-9]).

Task 2 – Task 2 firstly imports the Task1 as it contains Morse code dictionary then it takes the morse code input from the user. A user can enter as many sequences as it wishes. If the entered sequence contains anything other than 0,1 or *, the program will terminate immediately with the error message. If the user enters a blank string, the program will terminate.

Task3 – Task 3 first imports Task2 which recursively imports Task1. Morse code input given by the user in Task2 is converted to the respective character value which is present against it in the dictionary which was defined in Task1. If the morse code is not present in the dictionary then an error message is generated stating that the particular morse code is not present in the dictionary.

Task4 - Task 4 first imports Task3. In Task 4, we have designed a small analysis program. The objective of this program is to give us the occurrence of an individual character in a particular sequence and also the occurrence of individual characters in all the sequences.

Basic Assumptions which are required for successful execution of the program and understanding the result:

1. Before executing the program please download numpy package because to define array numpy package is being used. If numpy package is not installed on the pc it will throw an error and program will end abruptly.
2. To install numpy package, run anaconda navigator >> open jupyter notebook >> open new Terminal window and write "pip install numpy". Press Enter. Numpy package will get installed.
3. Task 4 should be executed if a full program needs to be executed. It will execute Task3, Task2, Task1 too.
4. In Task 2, after entering every morse code press * before entering new morse code in the sequence.
5. Please do not enter any values other than sequences of 0,1 and * else program will end immediately giving the error that invalid input and will only interpret the sequences which were entered before the invalid sequence.
6. Multiple * between two Morse codes will be treated as single *.
7. After entering a sequence press enter, a prompt will be generated which will ask if you want to enter
   another sequence or not.If you wish to enter another sequence then please press "y". Press "n" to stop giving input. As soon as you press "n", program moves to the Task3.
8. y" and "n" should be in lower case else subprogram will keep asking you to make choice.
9. Task1, Task2, Task3, and Task4 are python files with .py extension.

Python version used: Python 3.6.4

Software requirements :

1. Download Anaconda 3.6 version from https://www.anaconda.com/download/. It comes with built-in python3.6.
2. Other than Anaconda, Pycharm can also be used for running and writing python code. You can download Pycharm from https://www.jetbrains.com/pycharm/download/. Download the community version as its free.

Required python packages: Numpy

Running the Program :

1. First, download the folder A1 to your system and unzip it.
2. Now open Anaconda Navigator >> open jupyter notebook.
3. Home window will open, click on upload and navigate to folder A1. Select Task1.py, Task2.py, Task3.py and Task4.py files and click on okay.
4. Now click on upload which is in blue against the file.
5. If numpy package is not installed then open the terminal window and install numpy package.
6. To install, on the home page click on new >> Terminal. In terminal write “pip install numpy”. Enter. This will install the numpy package.
7. Now you can run the program in two ways :
   a. The first way is using Python3 file:
      i. First, open a new Python3 file.
      ii. Write %run Task4.py. Execute it.
   b. The second way is using terminal :
      i. Open new Terminal window.
      ii. Write python3.6.4 Task4.py.Enter

Notes :

1. Screenshots of the basic output of the program have been captured and saved in file “Example output of program”.