Morse

Write a Python program to encode and decode messages written in Morse alphabet. The Morse alphabet associates to each alphabetic, numerical and punctuation character a variable length code, made up of lines and points. The conversion table is contained in the file "*morse.txt*". The first lines of the file are show below:

```
  A .-
  B -...
  C -.-.
  D -..
  E .
  F ..-.
  G --.
  (etc.)
```

The first character on each line represents the standard alphabet character, and is followed by the corresponding Morse encoding. The two fields are separated by a white space.

The program must read a file *"commands.txt"* that contains the sequence of encoding / decoding operations to execute. The file is composed of multiple lines, and each line contains two fields, separated by a white space:

- the first field contains the character `e` for encoding operations (text to Morse), or character `d` for decoding operations (Morse to text)
- the second field is the name of the file that must be decoded or encoded. Each file will contain a single line of text.

The program must print on screen the translation (encoding or decoding) of all the required files.

NOTES

When encoding, the program must skip all characters that do not appear in the Morse alphabet (file *morse.txt*), and must not differentiate between lower and upper case characters. The program must separate with a space the Morse symbols printed on screen.

When decoding, assume that the input file contains Morse codes separated by spaces. Unrecognized codes (i.e., codes not appearing in the *morse.txt* file) must be ignored.

Example

Assuming a file *commands.txt* with the following contents:

```
e text.txt
d encoded.txt
```

with a file *text.txt* with contents

```
Hello world!
```

and a file *encoded.txt* with contents

```
.... . .-.. .-.. --- .-- --- .-. .-.. -..
```

the program output will be

```
Encoding file text.txt:
.... . .-.. .-.. --- .-- --- .-. .-.. -..
Decoding file encoded.txt:
HELLOWORLD
```