```cpp
//
//  main.cpp
//  A3_COSC_1437_014
//
//  Created by Nicholas Brown on 2/3/25.
//

#include <iostream>
#include <string>
using namespace std;


string toMorse( const char* morse[],  const char* characters[], const int SIZE, const string& input);


int main() {

    const int SIZE = 40;

    const char* morse[SIZE] = {"space","--..--",".-.-.-","..--..",
        ".-", "-...", "-.-.", "-..", ".", "..-.", "--.", "....", "..", ".---",
        "-.-", ".-..", "--", "-.", "---", ".--.", "--.-", ".-.", "...", "-",
        "..-", "...-", ".--", "-..-", "-.--", "--..", "-----", ".----", "..---",
        "...--", "....-", ".....", "-....", "--...", "---..", "----."
    };

    const char* characters[SIZE] = {" ", ",","."."?",
        "A", "B", "C", "D", "E", "F", "G", "H", "I", "J",
        "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T",
        "U", "V", "W", "X", "Y", "Z", "0", "1", "2", "3",
        "4", "5", "6", "7", "8", "9"
    };
    cout << "Please enter a string: " << endl;
    string input;

    getline(cin, input); // Read the entire input string (tolarates spaces)
    cout << "Translation: " << endl;
    cout << toMorse(morse, characters, SIZE, input) << endl;
    return 0;
}


// Breaking down the passing variables for review:
/*


const char*
 -Since we are using elements that are FIXED and shouldnt be messed with we use "const" we also use the "char*"
 since its easier to reference a value then copy it's information everywhere.
 -NULL terminated = use a pointer
```

```cpp
45
46
47   const char*
48   -Since we are using elements that are FIXED and shouldnt be messed with we use "const" we also use the "char*"
49   since its easier to reference a value then copy it's information everywhere.
50   -NULL terminated = use a pointer
51
52
53   const string&
54   -First off the const avoids anyone from messing with it's original value. The appersand next to the datatype tells us that we look at the variable directly (no copy)
55   -Since strings have their own built in NULL we can just pass by reference with '&' directly.
56   */
57
58   string toMorse(const char* morse[], const char* characters[], const int SIZE, const string& input) {
59       string result;
60
61       for (char ch : input) { // Essential for the given input be all Uppercased so that the translation is as accurate as possible.
62           ch = toupper(ch); // Convert to uppercase
63           int index = -1; // default index to -1 so that when there is an element it can start at 0 and pop up from there.. this also helps us with our later if statement below
64
65           // Find the index of the character in the `characters` array
66           for (int i = 0; i < SIZE; i++) {
67               if (characters[i][0] == ch) { // this looks like a 2D array but isnt.. this part of the problem tripped me up with syntax.
68                   index = i;
69               }
70           }
71           if(index != -1){
72               result += morse[index]; // Append the Morse code
73               result += " "; // Append some spaces for read-ability
74
75           }
76       }
77       return result;
78   }
79
80
```