



# Morse Code Encoder

Write a C++ program that converts a string of text into the equivalent Morse code.

Pictured to the right is the International version of Morse Code.

For the purposes of this exercise:

- Use "-" for dashes and "." for dots.
- Do not put spaces between dots and dashes within a letter.
- Separate letters with onespace.
- Separate words with three spaces.

Use `getline()` to get the user input.  
See "Strings and Chars" booklet for more information.

Example:

Input: `hello bob`

Output: `.... . .-.. .-.. --- -... --- ....`

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	--..
1	....-
2	..-.-
3	...--
4	....-
5	.....
6	-.....
7	--....
8	---...
9	----.
0	-----

# Solution

```
/* -----  
Morse Code Encoder  
by Mr. Giansante  
November 11, 2018  
  
This is the third way we solved this problem.  
  
It unnecessarily wastes memory by declaring an array that is larger  
than needed, but the processing required in each iteration of the loop  
is minimal compared to the previous two methods we explored.  
----- */  
  
#include <iostream>  
#include <string>  
#include <cctype>  
  
using namespace std;  
  
int main()  
{  
    string userInput;  
    string morsecode;  
  
    string morse[91];  
    int temp, temp2;  
  
    // Note: the index of the array corresponds directly to each  
    // character's ASCII code.  
  
    morse[32] = " "; // space  
  
    morse[48] = "-.... : // 0  
    morse[49] = ".---- : // 1  
    morse[50] = "-.--- : // 2  
    morse[51] = ".--.. : // 3  
    morse[52] = "-...- : // 4  
    morse[53] = ".---- : // 5  
    morse[54] = "-.... : // 6  
    morse[55] = "-...- : // 7  
    morse[56] = "-.... : // 8  
    morse[57] = "-.... : // 9  
  
    morse[65] = "-. : // A  
    morse[66] = "-... : // B  
    morse[67] = "-.-. : // C  
    morse[68] = "-.- : // D  
    morse[69] = ". : // E  
    morse[70] = "-.- : // F  
    morse[71] = "-- : // G  
    morse[72] = ".-.- : // H  
    morse[73] = ".- : // I  
    morse[74] = "-.-.- : // J  
    morse[75] = "-.- : // K  
    morse[76] = "-.-.- : // L  
    morse[77] = "--.- : // M  
    morse[78] = "-.- : // N  
    morse[79] = "-.-.- : // O  
    morse[80] = "-.-.- : // P  
    morse[81] = "--.- : // Q  
    morse[82] = ".-.- : // R  
    morse[83] = ".-.- : // S  
    morse[84] = "-.- : // T  
    morse[85] = "-.-.- : // U  
    morse[86] = "-.-.- : // V  
    morse[87] = "--.- : // W  
    morse[88] = "-.-.- : // X  
    morse[89] = "-.-.- : // Y  
    morse[90] = "-.-.- : // Z  
  
    // getline(cin, userInput);  
    userInput = "Morse Code SOS"; // only for testing  
  
    morsecode = "";  
  
    for(int i = 0; i < userInput.length(); i++)  
    {  
        morsecode = morsecode + morse[(int)toupper(userInput.at(i))];  
    }  
  
    cout << morsecode << "\n";  
  
    return 0;  
}
```