

Isomorphic Strings in C++																													
<pre>#include <iostream> #include <string> #include <unordered_map> using namespace std; bool iso(string s, string t) { if (s.length() != t.length()) { return false; } unordered_map<char, char> map1; // Maps characters from s to t unordered_map<char, bool> map2; // Tracks characters used in t for (int i = 0; i < s.length(); i++) { char ch1 = s[i]; char ch2 = t[i]; if (map1.count(ch1) > 0) { // If ch1 is already mapped if (map1[ch1] != ch2) { // Check if mapping is consistent return false; } } else { // ch1 has not been mapped yet if (map2.count(ch2) > 0) { // If ch2 is already mapped by another character in s return false; } else { // Create new mapping map1[ch1] = ch2; map2[ch2] = true; } } } return true; } int main() { string s1 = "abc"; string s2 = "cad"; cout << boolalpha << iso(s1, s2) << endl; // Output: true return 0; }</pre>	<h3>Step 1: Initialize Variables</h3> <ul style="list-style-type: none">• Input Strings: $s = \text{"abc"}, t = \text{"cad"}$• Maps Used:<ul style="list-style-type: none">○ $\text{map1} \rightarrow$ Stores mapping from s to t○ $\text{map2} \rightarrow$ Tracks characters already mapped in t <h3>Step 2: Iterating Through s and t</h3> <table><tr><th>Index (i)</th><th>s[i]</th><th>t[i]</th><th>map1 (s \rightarrow t)</th><th>map2 (used t characters)</th><th>Check for Conflict?</th><th>Result</th></tr><tr><td>0</td><td>'a'</td><td>'c'</td><td>{ a \rightarrow c }</td><td>{ c \rightarrow true }</td><td>No</td><td>Continue</td></tr><tr><td>1</td><td>'b'</td><td>'a'</td><td>{ a \rightarrow c, b \rightarrow a }</td><td>{ c \rightarrow true, a \rightarrow true }</td><td>No</td><td>Continue</td></tr><tr><td>2</td><td>'c'</td><td>'d'</td><td>{ a \rightarrow c, b \rightarrow a, c \rightarrow d }</td><td>{ c \rightarrow true, a \rightarrow true, d \rightarrow true }</td><td>No</td><td>Continue</td></tr></table> <h3>Step 3: Return Result</h3> <ul style="list-style-type: none">• Since no conflicts were found, return <code>true</code>. <h3>Final Output</h3> <p>true</p>	Index (i)	s[i]	t[i]	map1 (s \rightarrow t)	map2 (used t characters)	Check for Conflict?	Result	0	'a'	'c'	{ a \rightarrow c }	{ c \rightarrow true }	No	Continue	1	'b'	'a'	{ a \rightarrow c, b \rightarrow a }	{ c \rightarrow true, a \rightarrow true }	No	Continue	2	'c'	'd'	{ a \rightarrow c, b \rightarrow a, c \rightarrow d }	{ c \rightarrow true, a \rightarrow true, d \rightarrow true }	No	Continue
Index (i)	s[i]	t[i]	map1 (s \rightarrow t)	map2 (used t characters)	Check for Conflict?	Result																							
0	'a'	'c'	{ a \rightarrow c }	{ c \rightarrow true }	No	Continue																							
1	'b'	'a'	{ a \rightarrow c, b \rightarrow a }	{ c \rightarrow true, a \rightarrow true }	No	Continue																							
2	'c'	'd'	{ a \rightarrow c, b \rightarrow a, c \rightarrow d }	{ c \rightarrow true, a \rightarrow true, d \rightarrow true }	No	Continue																							
Output: true																													