Design an algorithm to encode **a list of strings** to **a string**. The encoded string is then sent over the network and is decoded back to the original list of strings.

Machine 1 (sender) has the function:

string encode(vector<string> strs) {

// ... your code

return encoded\_string;

}

Machine 2 (receiver) has the function:

vector<string> decode(string s) {

//... your code

return strs;

}

So Machine 1 does:

string encoded\_string = encode(strs);

and Machine 2 does:

vector<string> strs2 = decode(encoded\_string);

strs2 in Machine 2 should be the same as strs in Machine 1.

Implement the encode and decode methods.

You are not allowed to solve the problem using any serialize methods (such as eval).

**Example 1:**

**Input:** dummy\_input = ["Hello","World"]

**Output:** ["Hello","World"]

**Explanation:**

Machine 1:

Codec encoder = new Codec();

String msg = encoder.encode(strs);

Machine 1 ---msg---> Machine 2

Machine 2:

Codec decoder = new Codec();

String[] strs = decoder.decode(msg);

**Example 2:**

**Input:** dummy\_input = [""]

**Output:** [""]

**Constraints:**

* 1 <= strs.length <= 200
* 0 <= strs[i].length <= 200
* strs[i] contains any possible characters out of 256 valid ASCII characters.

**Follow up:**Could you write a generalized algorithm to work on any possible set of characters?