A screenshot of a cell phone

Description automatically generated

#include <iostream>

#include <string>

#include <vector>

using namespace std;

void recurse(vector<string> words, string joint\_string, vector<string> result) {

if(words.empty()) {

result.push\_back(joint\_string);

return;

}

// for(int i = 0 ; i < words.size(); ++i) {

for(auto ite = words.begin() ; ite < words.end(); ++ite) {

recurse(words.erase(ite), joint\_string.append(\*iter), result);

}

}

vector<int> find\_string(string str, string pattern, vector<int> & result) {

// vector<int> result;

if(pattern.size() > str.size()) return 0; // if wrong !!!!

for(int i = 0; i < str.size() - pattern.size(); ++i) {

if(str.at(i) == pattern.front()) {

// judge two string equal

if(judge\_two\_string\_equal(str+i, pattern));

result.push\_back(i);

}

}

return result;

}

bool judge\_two\_string\_equal(string str1, str2) {

int len1 = str1.size();

int len2 = str2.size();

int small = len1 > len2 ? len2 : len1;

for(int i = 0; i < small; ++i) {

if(str1.at(i) != str2.at(i)) {

return 0;

}

}

return 1;

}

int main() {

vector<string> words({"foo", "bar"});

string input("foobar");

// {foo, bar} ==> foobar, bar foo

// find\_all\_combination(vector<string> words);

vector<string> result;

recurse(words, string(), result);

// find all matched.

vector<int> pos;

for(auto str : result)

find\_string(input, str, pos);

for(auto i : pos) cout << i << endl;

return 0;

}