class Solution {

public:

std::vector<std::string> commonChars(std::vector<std::string>& A) {

auto vm{ std::vector<std::map<char, int>>() };

vm.resize(A.size());

for(size\_t i = 0; i < A.size(); i++)

{

for(const auto & el: A[i])

{

vm[i][el]++;

}

}

for(size\_t i = 0; i < A.size(); i++)

{

for(char c = 'a'; c <= 'z'; c++)

{

auto flg{ (bool) true };

for(size\_t k = 0; k < A[i].size(); k++)

{

if(c == A[i][k])

{

flg = false;

break;

}

}

if(flg)

{

vm[i][c] = 0;

}

}

}

auto res { std::map<char, int>() };

for(char c = 'a'; c <= 'z'; c++)

{

auto min\_loc{ (int) vm[0][c] };

for(size\_t i = 1; i < A.size(); i++)

{

if(min\_loc > vm[i][c])

{

min\_loc = vm[i][c];

}

}

res[c] = min\_loc;

}

auto res2 {std::vector<std::string>() };

for(const auto & el: res)

{

for(size\_t kk = 0; kk < el.second; kk++)

{

res2.push\_back(std::string(1, el.first));

}

}

return res2;

}

};