Aim-Given two integers n and k, return all possible combinations of k numbers chosen from the range [1, n}

#include <iostream>

#include <vector>

void backtrack(int start, int n, int k, std::vector<int>& combination, std::vector<std::vector<int>>& result) {

if (combination.size() == k) {

result.push\_back(combination);

return;

}

for (int i = start; i <= n; ++i) {

combination.push\_back(i);

backtrack(i + 1, n, k, combination, result);

combination.pop\_back(); // Backtrack

}

}

std::vector<std::vector<int>> combine(int n, int k) {

std::vector<std::vector<int>> result;

std::vector<int> combination;

backtrack(1, n, k, combination, result);

return result;

}

int main() {

int n, k;

std::cout << "Enter n: ";

std::cin >> n;

std::cout << "Enter k: ";

std::cin >> k;

std::vector<std::vector<int>> combinations = combine(n, k);

std::cout << "Combinations:\n";

for (const auto& comb : combinations) {

std::cout << "[ ";

for (int num : comb) {

std::cout << num << " ";

}

std::cout << "]\n";

}

return 0;

}