def find\_max\_and\_position():

numbers = [1101, 1111, 110101, 110111, 111101, 101111, 111111, 100010101, 100010111, 100011101, 100011111, 100101011, 100101101, 100101111, 100110101, 100110111, 100111011, 100111101, 100111111, 101010101, 101010111, 101011011, 101011111, 101101011, 101101101, 101101111, 101110101, 101110111, 101111011, 101111101, 101111111]

max\_num = 0

position = 0

for i, num in enumerate(numbers):

decimal\_num = int(num, 2)

if decimal\_num > max\_num and '11' not in num[1:]:

max\_num = decimal\_num

position = num.index('11') + 1 if '11' in num else 0

return max\_num, position

max\_num, position = find\_max\_and\_position()

print("The maximum number is", max\_num)

if position:

print("The series of two consecutive ones begins from the", str(position) + "th", "position.")

else:

print("There is no series of two consecutive ones.")

#include <iostream>

#include <vector>

using namespace std;

// Структура, представляющая товар

struct Product {

int id; // Идентификатор товара

int price; // Цена товара

int quantity; // Количество товара

Product(int \_id, int \_price, int \_quantity) : id(\_id), price(\_price), quantity(\_quantity) {}

};

// Функция для генерации всех возможных вариантов списков товаров скидок

void generateDiscountLists(vector<Product>& products, vector<int>& discountList, int currentIndex, int remainingQuantity) {

// Базовый случай: все товары в списке скидок уже выбраны

if (currentIndex == products.size()) {

// Выводим список товаров скидок

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

cout << products[i].id << " ";

}

cout << endl;

return;

}

// Рекурсивно генерируем все возможные комбинации

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

// Устанавливаем количество текущего товара в списке скидок

discountList[currentIndex] = i;

// Рекурсивно генерируем остальные товары в списке скидок

generateDiscountLists(products, discountList, currentIndex + 1, remainingQuantity - i);

}

}

int main() {

int numProducts; // Количество товаров

int maxQuantity; // Максимальное количество товаров для каждого артикула

// Вводим количество товаров и максимальное количество товаров для каждого артикула

cout << "Введите количество товаров: ";

cin >> numProducts;

cout << "Введите максимальное количество товаров для каждого артикула: ";

cin >> maxQuantity;

// Создаем вектор товаров

vector<Product> products;

for (int i = 1; i <= numProducts; i++) {

int price, quantity;

cout << "Введите цену товара " << i << ": ";

cin >> price;

cout << "Введите количество товара " << i << ": ";

cin >> quantity;

products.emplace\_back(i, price, quantity);

}

// Создаем вектор для хранения списка скидок

vector<int> discountList(numProducts, 0);

// Генерируем все возможные варианты списков товаров скидок

generateDiscountLists(products, discountList, 0, maxQuantity);

return 0;

}