#include <iostream>

#include <string>

using namespace std;

struct OrderDetails {

int numSacksCement = 0;

int numSacksGravel = 0;

int numSacksSand = 0;

int totalRejected = 0;

float totalWeight = 0;

};

// Function to check the contents and weight of a single sack (Task 1)

void checkSingleSack(OrderDetails &orderDetails) {

char contents;

float weight;

cout << "Enter the contents (C for cement, G for gravel, S for sand): ";

cin >> contents;

contents = toupper(contents);

cout << "Enter the weight of the sack in kilograms: ";

cin >> weight;

bool validSack = false;

string rejectionReasons = "";

// Check contents and weight of the sack

if ((contents == 'C' && weight > 24.9 && weight < 25.1) ||

((contents == 'G' || contents == 'S') && weight > 49.9 && weight < 50.1)) {

validSack = true;

orderDetails.totalWeight += weight;

} else {

if (contents != 'C' && contents != 'G' && contents != 'S') {

rejectionReasons += "Invalid contents. ";

}

if (!((contents == 'C' && weight > 24.9 && weight < 25.1) ||

((contents == 'G' || contents == 'S') && weight > 49.9 && weight < 50.1))) {

rejectionReasons += "Invalid weight for contents.";

}

orderDetails.totalRejected++;

}

// Output the result

if (validSack) {

cout << "Sack accepted! Contents: " << contents << ", Weight: " << weight << " kg" << endl;

} else {

cout << "Sack rejected. Reasons: " << rejectionReasons << endl;

}

}

// Function to check a customer’s order for delivery (Task 2)

void checkCustomerOrder(OrderDetails &orderDetails) {

cout << "Enter the number of cement sacks required: ";

cin >> orderDetails.numSacksCement;

cout << "Enter the number of gravel sacks required: ";

cin >> orderDetails.numSacksGravel;

cout << "Enter the number of sand sacks required: ";

cin >> orderDetails.numSacksSand;

for (int i = 0; i < orderDetails.numSacksCement + orderDetails.numSacksGravel + orderDetails.numSacksSand; ++i) {

cout << "Checking sack " << i + 1 << ":" << endl;

checkSingleSack(orderDetails);

}

// Output total weight and number of rejected sacks

cout << "Total weight of the order: " << orderDetails.totalWeight << " kg" << endl;

cout << "Number of sacks rejected from the order: " << orderDetails.totalRejected << endl;

}

// Function to calculate the price for a customer’s order (Task 3)

void calculateOrderPrice(OrderDetails &orderDetails) {

// Prices for sacks

int priceCement = 3;

int priceGravel = 2;

int priceSand = 2;

int specialPackPrice = 10;

cout << "Calculating the price for the order..." << endl;

checkCustomerOrder(orderDetails);

// Calculate regular price for the order

int regularPrice = (orderDetails.numSacksCement \* priceCement) +

(orderDetails.numSacksGravel \* priceGravel) +

(orderDetails.numSacksSand \* priceSand);

// Calculate number of special packs

int specialPacks = min(orderDetails.numSacksCement,

min(orderDetails.numSacksGravel / 2, orderDetails.numSacksSand / 2));

// Calculate discount price and amount saved

int discountPrice = specialPacks \* specialPackPrice;

int newPrice = regularPrice - discountPrice;

if (specialPacks > 0) {

cout << "Regular price for the order: $" << regularPrice << endl;

cout << "Discount price for special packs: $" << discountPrice << endl;

cout << "New price for the order: $" << newPrice << endl;

cout << "Amount saved: $" << discountPrice << endl;

} else {

cout << "Regular price for the order: $" << regularPrice << endl;

}

}

int main() {

// Test the functions for the tasks

OrderDetails order;

cout << "TASK 1 - Check the contents and weight of a single sack" << endl;

checkSingleSack(order);

cout << "\nTASK 2 - Check a customer’s order for delivery" << endl;

checkCustomerOrder(order);

cout << "\nTASK 3 - Calculate the price for a customer’s order" << endl;

calculateOrderPrice(order);

return 0;

}