#include<iostream>

using namespace std;

class Artikal {

char \*sifra;

char \*naziv;

double cijena;

int kolicinaNaSkladistu;

public:

//potrebne konstruktor i destruktor funkcije (imati na umu i konstruktor kopije)

//Operator -= na način da se stanje na skladištu artikla promijeni za proslijeđenu vrijednost

//Operator += na način da se stanje na skladištu artikla promijeni za proslijeđenu vrijednost

//Operator ++ na način da se cijena artikla uveća za 1

//Operator -- na način da se cijena artikla umanji za 1

//Operator == koji poredi artikle na osnovu sifre

//Preklopiti operatore za upis i ispis podataka o artiklu

int GetKolicinaNaSkladistu() { return kolicinaNaSkladistu; }

double GetCijena() { return cijena; }

char \*GetSifra() { return sifra; }

};

class StavkaRacuna {

Artikal artikal;

int \*kolicina;

public:

//potrebne konstruktor i destruktor funkcije (imati na umu i konstruktor kopije)

//operator dodjele =

//operator += koji uvećava stanje na skladištu artikla

//operator -= koji umanjuje stanje na skladištu artikla (jeste da u artiklu imamo ovo već preklopljeno, ali artikal je

//private atribut klase, ne možemo mu tek tako pristupati iz računa

//Operator poređenja== koji poredi stavke po sifri artikla

//operator ++ koji uvećava količinu stavke racuna

//operatore za ispis i unos podataka

double GetCijena() {//vraća cijenu stavke racuna (kolicina \* cijena artikla)

}

double GetKolicina() { //vraća količinu stavke računa }

};

int pom = 1000;

class Racun {

int brojRacuna;

StavkaRacuna stavke[100];

int brojStavki;

double iznosBezPDV;

double iznosSaPDV;

bool naplaceno;

public:

//potrebne konstruktor i destruktor funkcije (imati na umu i konstruktor kopije)

//operator dodjele =

//Operator() - dodaje nove artikle u stavke racuna (ulazni parametri su artikal i kolicina), kolicina nije obavezni parametar

//Voditi računa o tome da li je račun već naplaćen

//Ukoliko nema dovoljno artikala na stanju, ispisati odgovarajuću poruku

//Ukoliko artikal već postoji, potrebno je samo uvećati količinu za odgovarajuću stavku

//Prilikom svakog dodavanja artikla, azurirati vrijednosti iznosBezPDV i iznosSaPDV

//Operator -= uklanja artikal iz stavki na osnovu njegove sifre

//azurirati vrijednosti iznosBezPDV i iznosSaPDV

double PrintRacun() {

//Ispisuje sve stavke računa zajedno sa iznosom za plaćanje

//Omogućiti da se unese iznos koji je korisnik uplatio, te se

//vraća iznos kusura

}

};

void main() {

//main je moguće mijenjati i prilagođavati

//ali je ovo napisano sasvim dovoljno za testiranje funkcionalnosti zadatka

Racun racun;

Artikal artikli[3];

for (int i = 0; i < 3; i++)

{

cin >> artikli[i];

racun(artikli[i], 2);

}

//pokusaj dodavanja istog artikla na racun

racun(artikli[0], 2);

char sifra[20];

cout << "Unesite sifru artikla kojeg želite ukloniti: ";

cin.getline(sifra, 20);

racun -= sifra;

racun(artikli[1], 1);

cout<<"Kusur: "<<racun.PrintRacun()<<" KM"<<endl;

system("pause>0");

}