﻿using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

namespace Matricos1

{

class Matrica

{

const int MaxEil = 10;

const int MaxSt = 100;

private int[,] A;

public int n { get; set; }

public int m { get; set; }

public Matrica()

{

n = 0;

m = 0;

A = new int[MaxEil, MaxSt];

}

public void Dėti(int i, int j, int pirk)

{

A[i, j] = pirk;

}

public int ImtiReiksme(int i, int j)

{

return A[i, j];

}

}

class Program

{

const string duom = "..\\..\\duom.txt";

const string rez = "..\\..\\rez.txt";

static void Main(string[] args)

{

Matrica prekybosBaze = new Matricos1.Matrica();

Skaityti(duom, ref prekybosBaze);

if (File.Exists(rez))

File.Delete(rez);

Spausdinti(rez, prekybosBaze, " Pradiniai duomenys");

KiekvienaKasaAptarnavo(rez, prekybosBaze);

KiekvienąDienąAptarnauta(rez, prekybosBaze);

Console.WriteLine("Programa baigė darbą");

}

static double KiekVidutiniskai(int i, Matrica A)

{

double vidutiniskai = 0;

for (int j = 0; j < A.m; j++)

{

vidutiniskai += A.ImtiReiksme(i, j);

}

return vidutiniskai / A.m;

}

static int VisoAptarnauta(Matrica A)

{

int suma = 0;

for (int i = 0; i < A.n; i++)

for (int j = 0; j < A.m; j++)

suma = suma + A.ImtiReiksme(i, j);

return suma;

}

static void Skaityti(string fd, ref Matrica prekybosBaze)

{

int nn, mm, skaic;

string line;

using (StreamReader reader = new StreamReader(fd))

{

line = reader.ReadLine();

string[] parts;

nn = int.Parse(line);

line = reader.ReadLine();

mm = int.Parse(line);

prekybosBaze.n = nn;

prekybosBaze.m = mm;

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

{

line = reader.ReadLine();

parts = line.Split(';');

for (int j = 0; j < mm; j++)

{

skaic = int.Parse(parts[j]);

prekybosBaze.Dėti(i, j, skaic);

}

}

}

}

static int KiekNedirbo(int i, Matrica A)

{

int count = 0;

for (int j = 0; j < A.m; j++)

{

if (A.ImtiReiksme(i, j) == 0)

{

count++;

}

}

return count;

}

static int KasosNumerisMinPirkėjų(Matrica A, ref int minpirkeju)

{

minpirkeju = 0;

int min = int.MaxValue;

int nr = 0;

for (int i = 0; i < A.n; i++)

{

int suma = 0;

for (int j = 0; j < A.m; j++)

suma = suma + A.ImtiReiksme(i, j);

if (suma < min)

{

minpirkeju = suma;

min = suma;

nr = i + 1;

}

}

return nr;

}

static int KasosNumerisMaxPirkėjų(Matrica A, ref int pirkeju)

{

pirkeju = 0;

int max = 0;

int nr = 0;

for (int i = 0; i < A.n; i++)

{

int suma = 0;

for (int j = 0; j < A.m; j++)

suma = suma + A.ImtiReiksme(i, j);

if (suma > max)

{

pirkeju = suma;

max = suma;

nr = i + 1;

}

}

return nr;

}

static void KiekvienąDienąAptarnauta(string CFr, Matrica A)

{

using (var fr = File.AppendText(CFr))

{

fr.WriteLine();

for (int j = 0; j < A.m; j++)

{

int suma = 0;

for (int i = 0; i < A.n; i++)

suma = suma + A.ImtiReiksme(i, j);

fr.WriteLine(" Diena nr. {0}: aptarnauta klientų - {1}.", j + 1, suma);

}

}

}

static void KiekvienaKasaAptarnavo(string rez, Matrica A)

{

using (var fr = File.AppendText(rez))

{

for (int i = 0; i < A.n; i++)

{

int suma = 0;

for (int j = 0; j < A.m; j++)

{

suma = suma + A.ImtiReiksme(i, j);

}

fr.WriteLine(" Kasa nr. {0} aptarnavo {1} klientų.", i + 1, suma);

}

}

}

static void Spausdinti(string rez, Matrica prekybosBaze, string antraštė)

{

using (var fw = File.AppendText(rez))

{

fw.WriteLine(antraštė + '\n');

fw.WriteLine(" Kasų kiekis {0}", prekybosBaze.n);

fw.WriteLine(" Darbo dienų kiekis {0}", prekybosBaze.m);

fw.WriteLine(" Aptarnautų klientų kiekiai:");

for (int i = 0; i < prekybosBaze.n; i++)

{

for (int j = 0; j < prekybosBaze.m; j++)

{

fw.Write("{0,4:d}", prekybosBaze.ImtiReiksme(i, j));

}

fw.WriteLine();

}

fw.WriteLine();

fw.WriteLine(" Rezultatai");

fw.WriteLine();

fw.WriteLine(" Viso aptarnauta: {0} klientų.", VisoAptarnauta(prekybosBaze));

fw.WriteLine();

int maxpirkeju = 0;

fw.WriteLine(" Daugiausia pirkėjų aptarnavo (kasa): {0}; aptarnavo {1} pirkėjų.", KasosNumerisMaxPirkėjų(prekybosBaze, ref maxpirkeju), maxpirkeju);

fw.WriteLine();

int minpirkeju = 0;

fw.WriteLine(" Mažiausia pirkėjų aptarnavo (kasa): {0}; aptarnavo {1} pirkėjų.", KasosNumerisMinPirkėjų(prekybosBaze, ref minpirkeju), minpirkeju);

fw.WriteLine();

for (int i = 0; i < prekybosBaze.n; i++)

{

fw.WriteLine(" Vidutiniškai per dieną kasą aptarnaudavo: {0, 5:f} klientų.", KiekVidutiniskai(i, prekybosBaze));

}

fw.WriteLine();

for (int i = 0; i < prekybosBaze.n; i++)

{

fw.WriteLine(" Kasa nedirbo: {0, 5:d} dienų.", KiekNedirbo(i, prekybosBaze));

}

fw.WriteLine();

}

}

}

}