//p62 ex35

int num,max=0,max\_num=0,min=1000000000,min\_num=0 ;

int[] arr = new int[15];

Console.WriteLine("enter num between1-15 to finish enter the number 0");

num = int.Parse(Console.ReadLine());

while (num!=0)

{

arr[num - 1]++;

Console.WriteLine("enter num between1-15 to finish enter the number 0");

num = int.Parse(Console.ReadLine());

}

for (int i = 0; i < arr.Length; i++)

{

Console.WriteLine(arr[i]);

if (min>arr[i])

{

min = arr[i];

min\_num = i + 1;

}

}

Console.WriteLine("min num "+min\_num);

Console.WriteLine("max num"+max\_num);

//63 ex38

int[] par = new int[50];

for (int i = 0; i < par.Length; i++)

{

par[i] = 0;

}

int[] judges = new int[3];

for (int i = 0; i <judges.Length; i++)

{

judges[i] = 0;

}

for (int i = 0; i < 1; i++)

{

Console.WriteLine("Nest 1 place");

judges[0] = int.Parse(Console.ReadLine());

Console.WriteLine("Nest 2 place");

judges[1] = int.Parse(Console.ReadLine());

Console.WriteLine("Nest 3 place");

judges[2] = int.Parse(Console.ReadLine());

par[judges[0]] += 3;

par[judges[1]] += 2;

par[judges[2]] += 1;

}

int max1 = 0,sav=0, max2 = 0,sav2=0, max3 = 0,sav3=0;

for (int i = 0; i < par.Length; i++)

{

if (max1<par[i])

{

max1 = par[i];

sav = i + 1;

}

}

for (int i = 0; i < par.Length; i++)

{

if (max2 < par[i]&&par[i]<max1)

{

max2 = par[i];

sav2 = i + 1;

}

}

for (int i = 0; i < par.Length; i++)

{

if (max3 < par[i] && par[i] < max2)

{

max3 = par[i];

sav3 = i + 1;

}

}

Console.WriteLine("1: "+ sav+ " Points " +max1);

Console.WriteLine("2: " + sav2 + " Points " + max2);

Console.WriteLine("3: " + sav3 + " Points " + max3);

namespace p77\_ex49

{

class Doctor

{

private string name;//שם

private int id;//תעודת זהות

private string specialization;//התחןת

private int several\_ranks ;//מספר מדרגים

private int average\_rating;//ממוצה דרוג

private string area\_in\_the\_country;//אזור בארץ

public Doctor(string name, int id, string specialization, string area\_in\_the\_country)

{

this.name = name;

this.id = id;

this.specialization = specialization;

this.area\_in\_the\_country = area\_in\_the\_country;

this.average\_rating = 0;

this.several\_ranks = 0;

}

public string GetName()

{

return this.name;

}

public int GetId()

{

return this.id;

}

public string GetSpecialization()

{

return this.specialization;

}

public int GetSeveral\_ranks()

{

return this.several\_ranks;

}

public int GetAverage\_rating()

{

return this.average\_rating;

}

public string GetArea\_in\_the\_country()

{

return this.area\_in\_the\_country;

}

public void SetName(string neamToSet)

{

this.name = neamToSet;

}

public void SetSpecialization(string specializationToSet)

{

this.specialization = specializationToSet;

}

public void SetAverage\_rating(int average\_ratingToSet)

{

this.average\_rating \*= this.several\_ranks;

this.several\_ranks++;

this.average\_rating += average\_ratingToSet;

this.average\_rating /= this.several\_ranks;

}

public override string ToString()

{

return this.name + "," + this. id+ "," + this.specialization+","+this.several\_ranks+","+this.average\_rating+","+ this.area\_in\_the\_country;

}

}

}

//p77 ex49

public static void Ex49(int n)

{

//רופים פעולה תקלותה האזור רפים באזור ואת את התמחןת זו שהדרוג N שלם ובונה מערך בגודל N הפונקצה מקבלת

Doctor[] doctors = new Doctor[n];

for (int i = 0; i < doctors.Length; i++)

{

Console.WriteLine("enetr name and id and specialization and area in the country");

doctors[i] = new Doctor(Console.ReadLine(), int.Parse(Console.ReadLine()), Console.ReadLine(), Console.ReadLine());

}

int id;

Console.WriteLine("enetr id To finish after that num 0");

id = int.Parse(Console.ReadLine());

while (id!=000)

{

for (int i = 0; i < doctors.Length; i++)

{

if (doctors[i].GetId()==id)

{

Console.WriteLine("enetr raising a doctor");

doctors[i].SetAverage\_rating(int.Parse(Console.ReadLine()));

}

}

Console.WriteLine("enetr id To finish after that num 0");

id = int.Parse(Console.ReadLine());

}

string area;

Console.WriteLine("The introduction of an area in Israel");

area = Console.ReadLine();

for (int i = 0; i < doctors.Length; i++)

{

if (doctors[i].GetArea\_in\_the\_country()==area)

{

Console.WriteLine(doctors[i]);

}

}

}

static void Main(string[] args)

{

//p77 ex49

Ex49(1);

}

//ex1

public static bool Lines(int[,]arr)

{

// הפעולה מקבלת מערך דו ממדי ומחזר אמת הם הוא ממוינת שורות שקר החרת

for (int i = 0; i < arr.GetLength(0); i++)

{

for (int j = 0; j < arr.GetLength(1); j++)

{

if (arr[i,j-1]>arr[i,j])

{

return false;

}

}

}

return true;

}

//ex1

public static bool Columns(int[,] arr)

{

// הפעולה מקבלת מערך דו ממדי ומחזר אמת הם הוא ממוינת עמודות שקר החרת

for (int j = 0; j < arr.GetLength(1); j++)

{

for (int i = 0; i < arr.GetLength(0); i++)

{

if (arr[i-1, j] > arr[i, j])

{

return false;

}

}

}

return true;

}

//ex1 mian

int[,] matrix = new int[10,15];

for (int i = 0; i < matrix.GetLength(0); i++)

{

for (int j = 0; j <matrix.GetLength(1); j++)

{

Console.WriteLine("enter num");

matrix[i, j] = int.Parse(Console.ReadLine());

}

}

if (Lines(matrix)==true&&Columns(matrix)==true)

Console.WriteLine("Fully sorted");

if(Lines(matrix) == true && Columns(matrix) == false)

Console.WriteLine("Sorted by rows");

if (Lines(matrix) == false && Columns(matrix) == true)

Console.WriteLine("Sorted columns");

//ex5

int[,] number = new int[10, 10];

Random r = new Random();

for (int i = 0; i < number.GetLength(0); i++)

{

for (int j = 0; j < number.GetLength(1); j++)

{

number[i, j] = r.Next(1,51);

}

}

int[,] arr = new int[2, 2];

for(int i = 0; i < arr.GetLength(0); i++)

{

for (int j = 0; j < arr.GetLength(1); j++)

{

Console.WriteLine("enter num");

arr[i, j]=int.Parse(Console.ReadLine());

}

}

for (int i = 0; i < number.GetLength(0); i++)

{

for (int j = 0; j < number.GetLength(1); j++)

{

Console.WriteLine(number[i,j]);

}

}

for (int i = 0; i < arr.GetLength(0); i++)

{

for (int j = 0; j < arr.GetLength(1); j++)

{

Console.WriteLine(arr[i, j] );

}

}

for (int i = 0; i < number.GetLength(0); i++)

{

for (int j = 0; j < number.GetLength(1); j++)

{

if (number[i,j]==arr[0,0]&&number[i+1,j]==arr[1,0]&&number[i,j+1]==arr[0,1]&&number[i+1,j+1]==arr[1,1])

{

Console.WriteLine("The big set decides the little brother too");

}

}

}

class Advr

{

private string advertisement\_name;//שם הפרסןמת

private string company\_name;//שם חברה

private int length;//אורך הפרםומת בשניות

public Advr(string advertisement\_name, string company\_name, int length)

{

this.advertisement\_name = advertisement\_name;

this.company\_name = company\_name;

this.length = length;

}

public string GetAdvertisement\_name()

{

return this.advertisement\_name;

}

public string GetCompany\_name()

{

return this.company\_name;

}

public int GetLength()

{

return this.length;

}

public void SetAdvertisement\_name(string advertisement\_nameToSet)

{

this.advertisement\_name = advertisement\_nameToSet;

}

public void SetCompany\_name(string company\_nameToSet)

{

this.company\_name = company\_nameToSet;

}

public void SetLength(int lengthToSet)

{

this.length = lengthToSet;

}

public override string ToString()

{

return this.advertisement\_name + "," + this. company\_name+ "," + this.length;

}

}

//p77 ex50 ב

public static int Ex50 (Advr[]arr ,string company\_name)

{

int time=0;

for (int i = 0; i < arr.Length; i++)

{

if (arr[i].GetCompany\_name()==company\_name)

{

time += arr[i].GetLength();

}

}

return time;

}

static void Main(string[] args)

{

//int second\_price;//מחיר לשנייה 800

Advr[] arr = new Advr[53];

for (int i = 0; i < arr.Length; i++)

{

Console.WriteLine("enter advertisement\_name and company\_name and length In seconds");

arr[i] = new Advr(Console.ReadLine(), Console.ReadLine(), int.Parse(Console.ReadLine()));

}

//arr[0] = new Advr("h", "h", 100);

int subtlety, Seconds;

for (int i = 0; i < arr.Length; i++)

{

subtlety =arr[i].GetLength()/60;

Seconds = arr[i].GetLength() - (subtlety \* 60);

Console.WriteLine("subtlety "+ subtlety+ " Seconds "+Seconds );

}

int sum\_advertising = 0, count\_company = 0;

for (int i = 0; i < arr.Length; i++)

{

Console.WriteLine("Cost of advertising"+arr[i].GetLength()\*800);

sum\_advertising += arr[i].GetLength() \* 800;

}

string name="";

Console.WriteLine("Cost per day"+sum\_advertising);

for (int i = 0; i < arr.Length; i++)

{

if (Ex50(arr,arr[i].GetCompany\_name())>count\_company)

{

count\_company = Ex50(arr, arr[i].GetCompany\_name());

name= arr[i].GetCompany\_name();

}

}

Console.WriteLine("name " + name+ " Revenue "+count\_company);

}

//ex6 א

public static bool Matrix\_thin(int[,] arr)

{

//פעולה מקבלת מערך דו מדי ומחזירה אמת כאשר המטצה דלילה

int count = 0, all\_count = 0;

for (int i = 0; i < arr.GetLength(0); i++)

{

for (int j = 0; j < arr.GetLength(1); j++)

{

if (arr[i, j] != 0)

count++;

else

all\_count++;

}

if (count > 2)

return false;

count = 0;

}

if (arr.Length \* 0.2 > all\_count)

return false;

return true;

}

//ex6 ב

public static int[,] Matrix (int[]values,int[]lines,int[]columns,int sum)

{

//פעולה מקבלת 3 מערכים ואת מספר איברים מטריצה מקורית ומחזירה את מטריצה המקורית

int[,] arr = new int[(int)Math.Sqrt(sum), (int)Math.Sqrt(sum)];

for (int i = 0; i < arr.GetLength(0); i++)

{

for (int j = 0; j < arr.GetLength(1); j++)

{

arr[i, j] = 0;

}

}

for (int i = 0; i < values.Length; i++)

{

arr[lines[i], columns[i]] = values[i];

}

return arr;

}

//ex6

int[,] number = new int[5,5];

Random r = new Random();

for (int i = 0; i < number.GetLength(0); i++)

{

for (int j = 0; j < number.GetLength(1); j++)

{

number[i, j] = r.Next(0, 10);

}

}

Console.WriteLine(Matrix\_thin(number));

int[] vlues = { 9, 8, 25, 15 };

int[] com = { 3, 2, 2, 4 };

int[] lien = { 0, 2, 3, 4 };

number =Matrix(vlues,lien,com,25);

for (int i = 0; i < number.GetLength(0); i++)

{

for (int j = 0; j < number.GetLength(1); j++)

{

Console.WriteLine( number[i, j] );

}

}

//ex11

int[,] number = new int[5, 4];

Random r = new Random();

for (int i = 0; i < number.GetLength(0); i++)

{

for (int j = 0; j < number.GetLength(1); j++)

{

number[i, j] = r.Next(0, 100);

}

}

int count\_column=0, count\_length=0;

for (int i = 0; i < number.GetLength(0); i++)

{

for (int j = 1; j < number.GetLength(1); j+=2)

{

count\_column+= number[i, j];

}

}

for (int i = 0; i < number.GetLength(0); i+=2)

{

for (int j = 1; j < number.GetLength(1); j ++)

{

count\_length += number[i, j];

}

}

if (count\_column>count\_length)

{

Console.WriteLine("Matrix postiive");

}

else

{

Console.WriteLine("Matrix not postiive");

}

class CoupleNames

{

private string name1;

private string name2;

public CoupleNames(string name1, string name2)

{

this.name1 = name1;

this.name2 = name2;

}

public string GetName1()

{

return this.name1;

}

public string GetName2()

{

return this.name2;

}

public void SetName1(string name1ToSet)

{

this.name1 = name1ToSet;

}

public void SetName2(string name2ToSet)

{

this.name2 = name2ToSet;

}

public int CompareTo(CoupleNames[,] arr)

{

int count=0;

for (int i = 0; i <=2&&arr.GetLength(0)>i ; i++)

{

for (int j = 0; j<=2&&j < arr.GetLength(1); j++)

{

count += 2;

}

}

return count;

}

public override string ToString()

{

return this.name1 + " , " + this.name2;

}

}

//p116 ex19 ג

public static int Num(CoupleNames[,] arr)

{

// ןמחזירה את מספר תהתלמידים בכיתה CoupleNames פעולה מקבלת מערך דו מדדי של טיפוס

int count = 0;

for (int i = 0; i < arr.GetLength(0); i++)

{

for (int j = 0; j < arr.GetLength(1); j++)

{

if (arr[i, j].GetName1() != "" && arr[i, j].GetName2() != "")

{

count += 2;

}

else if (arr[i, j].GetName1() == "" && arr[i, j].GetName2() != "")

{

count += 1;

}

else if (arr[i, j].GetName1() != "" && arr[i, j].GetName2() != "")

{

count += 1;

}

}

}

return count;

}

//p116 ex19 ד

public static int Location(CoupleNames[,] arr,string name)

{

// ושם תלמידה ומחירה מספר דו ספרתי מאתר את מקום CoupleNames פעולה מקבלת מערך דו מדדי של טיפוס

int num=0;

for (int i = 0; i < arr.GetLength(0); i++)

{

for (int j = 0; j < arr.GetLength(1); j++)

{

if (arr[i, j].GetName1()==name|| arr[i,j].GetName2()==name)

{

num = j \* 10 + i;

}

}

return num;

}

//p116 ex19

CoupleNames[,] cles = new CoupleNames[8, 4];

for (int i = 0; i < cles.GetLength(0); i++)

{

for (int j = 0; j < cles.GetLength(1); j++)

{

Console.WriteLine("enter name");

cles[i, j] = new CoupleNames(Console.ReadLine(), Console.ReadLine());

}

}

Console.WriteLine(Num(cles));

Console.WriteLine(Location(cles,"niv"));

}

class Trarin

{

private Engine engine; //קטר

private Carriage[]carriage; //מערך קרונות

private int num\_carriage; // מספר הקרונות שיש ברכבת בפועל

public Trarin(Engine engine, int N, int num\_carriage)

{

this.engine = engine;

this. num\_carriage= num\_carriage;

this.carriage = new Carriage[N];

}

public SetEngine(Engine engineToSet)

{

this.engine = engineToSet;

}

public bool Carriage\_pls(Carriage c)

{

for (int i = 0; i < this.carriage.Length; i++)

{

if (this.carriage[i]==null)

{

this.carriage[i] = c;

this.num\_carriage++;

return true;

}

}

return false;

}

public int Max()

{

int sum = 0;

for (int i = 0; i < this.carriage.Length; i++)

{

sum += this.carriage[i].GetMax\_Expriments();

}

return sum;

}

public int Avg\_Expriments ()

{

return Max /num\_carriage;

}

//p172 ex3

Trarin t = new Trarin(new Engine(5, 1900), 5, 4);

Engine e = new Engine(4, 1950);

t.SetEngine(e);

Carriage c=new Carriage(5,100,50)

t.Carriage\_pls(c);

Console.WriteLine(t.Max()); Console.WriteLine(t.Avg\_Expriments());

class Cube

{

private int length\_side; //אורך צלע של קובייה

private string color; //צבע הקובייה

public Cube(int length\_side, string color)

{

this.length\_side = length\_side;

this.color = color;

}

public int GetLength\_side()

{

return this.length\_side;

}

public string GetColor()

{

return this.color;

}

public void SetLength\_side(int length\_side)

{

this.length\_side = length\_side;

}

public void SetColor(string color)

{

this.color = color;

}

public bool Equals(Cube c)

{

if (c.GetColor()==this.color && c.GetLength\_side()==this.length\_side)

{

return true;

}

return false;

}

public override string ToString()

{

return this.length\_side +" "+ this.color;

}

}

class CubesTower

{

private int max\_cube;// מספר הקוביות מקסמלי שגדל יכל להכיל

private int num\_containing;//מספר הקוביות שמדל מסוים מכיל בפועל

private Cube[] collection\_of\_dice;// אוסף הקובית המונחות אחת על השנייה

public CubesTower(int max\_cube, int num\_containing, Cube[]collection\_of\_dice)

{

this.max\_cube = max\_cube;

this.num\_containing = num\_containing;

for (int i = 0; i < this.max\_cube; i++)

{

this.collection\_of\_dice[i] = collection\_of\_dice[i];

}

}

public int Tower\_at\_maximum\_capacity()

{

while (this.max\_cube > this.num\_containing)

{

this.num\_containing++;

}

return this.num\_containing;

}

public int Cube\_removal()

{

if (this.num\_containing<1)

{

this.num\_containing--;

}

return this.num\_containing;

}

public bool Color(string color)

{

for (int i = 0; i < this.collection\_of\_dice.Length; i++)

{

if (this.collection\_of\_dice[i].GetColor()==color)

{

return true;

}

}

return false;

}

public bool An\_empty\_tower()

{

if (this.num\_containing==0)

{

return true;

}

return false;

}

public bool Maximum\_occupancy()

{

if (this.num\_containing == this.max\_cube)

{

return true;

}

return false;

}

public int GetMax\_cube()

{

return this.max\_cube;

}

public int GetNum\_containing()

{

return this.num\_containing;

}

public Cube GetCollection\_of\_dice(int num)

{

return this.collection\_of\_dice[num];

}

public void SetMax\_cube(int max\_cube)

{

this.max\_cube = max\_cube;

}

public void SetNum\_containing(int num\_containin)

{

this.num\_containing = num\_containin;

}

public void SetCollection\_of\_dice(Cube c)

{

for (int i = 0; i < this.collection\_of\_dice.Length; i++)

{

this.collection\_of\_dice[i] = c[i];

}

}

public bool Equals(CubesTower c)

{

bool b = true;

if (c.GetMax\_cube()==this.max\_cube&&c.GetNum\_containing()==this.num\_containing)

{

for (int i = 0; i <this.collection\_of\_dice.Length; i++)

{

if (this.collection\_of\_dice[i]!=c.GetCollection\_of\_dice(i))

{

return false;

}

}

return true;

}

}

public override string ToString()

{

return this.max\_cube + "," + this.num\_containing ;

}

public string CubesTower\_color()

{

int sev = this.collection\_of\_dice.Length;

Cube[]arr= new Cube[max\_cube];

arr[0] =new Cube (this.collection\_of\_dice[0].GetLength\_side(), this.collection\_of\_dice[0].GetColor());

CubesTower c = new CubesTower(5,1,arr);

for (int i = 1; i <sev ; i++)

{

if (c.Color(this.collection\_of\_dice[i].GetColor())==false)

{

arr[i] = new Cube(this.collection\_of\_dice[i].GetLength\_side(), this.collection\_of\_dice[i].GetColor());

c.Tower\_at\_maximum\_capacity();

}

this.Cube\_removal();

}

return c.ToString();

}

}

static void Main(string[] args)

{

//p173\_ex5

Cube[] cub= new Cube [(2, "rad"),(2,"y")];

CubesTower c = new CubesTower(5, 2, cub) ;

Console.WriteLine("enetr -1 '+'Cube enetr '-'Cube enter 0 to dan");

int num =int.Parse( Console.ReadLine());

while (num!=0 && c.Maximum\_occupancy()== false)

{

if (num!=-1&&num!=-2)

{

Console.WriteLine("num not good");

}

if (num==-1)

{

c.SetNum\_containing(c.GetNum\_containing()+1);

}

if (num==-2)

{

c.SetNum\_containing(c.GetNum\_containing() - 1);

}

num = int.Parse(Console.ReadLine());

}

if (c.Maximum\_occupancy() == false)

{

Console.WriteLine("not in to mara");

}

Console.WriteLine(c.CubesTower\_color());

}

class Adress

{

private string street;

private int number;

private string city;

public Adress(string street, int number, string city)

{

this.street = street;

this.number = number;

this.city = city;

}

public string GetStreet()

{

return street;

}

public string GetCity()

{

return city;

}

public int GetNumber()

{

return number;

}

public void SetStreet(string street)

{

this.street = street;

}

public void SetCity(string city)

{

this.city = city;

}

public void SetNumber(int number)

{

this.number = number;

}

public override string ToString()

{

return "street: " + street + " number: " + number + " city:" + city;

}

}

class Room

{

private string type;

private int length;//אורך החדר

private int rorb;//רורב חדר

public Room(string type,int length, int rorb)

{

this.type = type;

this.length = length;

this.rorb = rorb;

}

public Room(Room room)

{

type = room.GetRoomType();

length = room.GetLength();

rorb = room.GetRorb();

}

public string GetRoomType()

{

return type;

}

public int GetLength()

{

return this.length;

}

public int GetRorb()

{

return this.rorb;

}

public void SetLength(int lengthToSet)

{

this.length = lengthToSet;

}

public void SetRorb(int rorbToSet)

{

this.rorb = rorbToSet;

}

public void SetType(string type)

{

this.type = type;

}

public void Copying\_Room(Room r)

{

this.length = r.GetLength();

this.rorb = r.GetRorb();

}

public int Area()

{

return this.length \*this.rorb;

}

public override string ToString()

{

return this.type+","+this.length + "," + this.rorb ;

}

}

class Apartment

{

private string name\_the\_landlord;//שם בעל הדירה

private Room[] rooms\_the\_apartment=new Room[10];//אוסף החדרים של הדירה

public Apartment(Room[] rooms\_the\_apartment, string name\_the\_landlord)

{

for (int i = 0; i < rooms\_the\_apartment.Length; i++)

{

if (rooms\_the\_apartment != null)

{

this.rooms\_the\_apartment[i] = new Room(rooms\_the\_apartment [i]);

}

}

this.name\_the\_landlord = name\_the\_landlord;

}

public Apartment(Apartment apartment)

{

rooms\_the\_apartment = apartment.GetRooms\_the\_apartment();

name\_the\_landlord = apartment.GetName\_the\_landlord();

}

public Room[] GetRooms\_the\_apartment()

{

return rooms\_the\_apartment;

}

public string GetName\_the\_landlord()

{

return name\_the\_landlord;

}

public void SetRooms\_the\_apartment(Room[] rooms\_the\_apartment)

{

this.rooms\_the\_apartment = rooms\_the\_apartment;

}

public void SetName\_the\_landlord(string name\_the\_landlord)

{

this.name\_the\_landlord = name\_the\_landlord;

}

public int GetApartmentArea()

{

int sum = 0;

for (int i = 0; i <this.rooms\_the\_apartment.Length; i++)

{

if (this.rooms\_the\_apartment[i] != null)

{

sum += this.rooms\_the\_apartment[i].Area();

}

}

return sum;

}

public string GetCategory()

{

if (GetApartmentArea() > 110)

return "large";

else if (GetApartmentArea() <= 110 && GetApartmentArea() > 70)

return "medium";

else

{

return "small";

}

}

public override string ToString()

{

string str = "";

for (int i = 0; i < rooms\_the\_apartment.Length; i++)

{

if (rooms\_the\_apartment[i] != null)

{

str += rooms\_the\_apartment[i].ToString() + " ";

}

}

return "rooms: " + str + "name\_the\_landlord: " + name\_the\_landlord;

}

}

class Building

{

private Adress adress;

private Apartment[] apartments = new Apartment[100];

public Building(Adress adress, Apartment[] apartments)

{

for (int i = 0; i < apartments.Length; i++)

{

if (apartments[i] != null)

{

apartments[i] = new Apartment(apartments[i]);

}

}

this.adress = new Adress(adress.GetStreet(), adress.GetNumber(), adress.GetCity());

}

public Adress GetAdress()

{

return adress;

}

public Apartment[] GetApartments()

{

return apartments;

}

public void SetApartments(Apartment[] apartments)

{

for (int i = 0; i < apartments.Length; i++)

{

if (apartments[i] != null)

{

apartments[i] = new Apartment(apartments[i]);

}

}

}

public void SetAdress(Adress adress)

{

this.adress = adress;

}

public override string ToString()

{

string str = "";

for (int i = 0; i < apartments.Length; i++)

{

str += apartments[i].ToString() + " ";

}

**return "apartments: " + str + " adress: " + adress.ToString();**

}

}

class Program

{

public static int GetApartmentArea(Apartment apartment)

{

int sum = 0;

for (int i = 0; i < apartment.GetRooms\_the\_apartment().Length; i++)

{

if (apartment.GetRooms\_the\_apartment()[i] != null)

{

sum += apartment.GetRooms\_the\_apartment()[i].Area();

}

}

return sum;

}

public static int GetAmountOfLarge(Building building)

{

int sum = 0;

for (int i = 0; i < building.GetApartments().Length; i++)

{

if (building.GetApartments()[i].GetCategory() == "large")

{

sum++;

}

}

return sum;

}

public static int GetMax(Building[] buildings)

{

int max = 0;

for (int i = 0; i < buildings.Length; i++)

{

if (max < GetAmountOfLarge(buildings[i]))

{

max = GetAmountOfLarge(buildings[i]);

}

}

return max;

}

public static void PrintMostLarge(Building[] buildings)

{

int max = GetMax(buildings);

for (int i = 0; i < buildings.Length; i++)

{

if (GetAmountOfLarge(buildings[i]) == max)

{

Console.WriteLine(buildings);

}

}

}

static void Main(string[] args)

{

//p 174 ex6

Room[] rooms = { new Room("hi", 12, 45), new Room("hi", 12, 45), new Room("hi", 54, 45), new Room("hi", 11, 6), new Room("hi", 12, 89), new Room("hi", 46, 12), new Room("hi", 5, 78) };

Room[] rooms2 = { new Room("hi", 145, 456), new Room("hi", 456, 45646), new Room("hi", 54, 456456), new Room("hi", 11, 456), new Room("hi", 12, 789), new Room("hi", 46, 7), new Room("hi", 5, 987) };

Apartment[] apartments = { new Apartment(rooms, "yotam"), new Apartment(rooms2, "yotam") };

Adress adress = new Adress("yotam", 12, "ramat hasharon");

Adress adress1 = new Adress("yotam", 45, "hi");

Room[] rooms3 = { new Room("hi", 12, 456456), new Room("hi", 12, 456546), new Room("hi", 54, 456456), new Room("hi", 11, 6), new Room("hi", 12, 456456), new Room("hi", 456456, 12), new Room("hi", 456, 78) };

Room[] rooms4 = { new Room("hi", 145, 456), new Room("hi", 456, 45646), new Room("hi", 54, 456456), new Room("hi", 11, 456), new Room("hi", 12, 789), new Room("hi", 46, 7), new Room("hi", 5, 987) };

Apartment[] apartments2 = { new Apartment(rooms3, "yotam"), new Apartment(rooms4, "yotam") };

Building[] buildings = { new Building(adress, apartments), new Building(adress1, apartments2) };

}

}

class Id\_Adult

{

private string id\_num;//מספר תעודת זהות

private string first\_name;//שם פרטי

private string last\_name;//שם משפחה

private Date birth;//תאריך לידה

private string country\_of\_birth;//ארץ לידה

private Id\_Kid[] kids;//תעודות זהות של ילדיו

public Id\_Adult(string id\_num, string first\_name, string last\_name, Date birth, string country\_of\_birth,int num\_kids, Id\_Kid[] kids)

{

this.id\_num = id\_num;

this.first\_name = first\_name;

this.last\_name = last\_name;

this.birth = birth;

this.country\_of\_birth = country\_of\_birth;

this.kids = new Id\_Kid[num\_kids];

for (int i = 0; i <kids.Length; i++)

{

this.kids[i] = kids[i];

}

}

public Id\_Kid[] GetKids()

{

return this.kids;

}

public bool Add\_kid(Id\_Kid id)

{

for (int i = 0; i < this.kids.Length; i++)

{

if (null==this.kids[i])

{

this.kids[i] = id;

return true;

}

}

return false;

}

public void Eightiiln(Id\_Kid id)

{

for (int i = 0; i < this.kids.Length; i++)

{

if (this.kids[i]!=null&& id!=null && this.kids[i].GetId\_num() == id.GetId\_num())

{

this.kids[i] = null;

}

}

}

public int Adults(Id\_Adult id)

{

int count = 0;

for (int i = 0; i < this.kids.Length; i++)

{

for (int j = 0; j < id.GetKids().Length; j++)

{

if (this.kids[i] != null && id.GetKids()[j] != null && id.GetKids()[j].GetId\_num()==this.kids[i].GetId\_num())

{

count++;

}

}

}

return count;

}

public string GetFirst\_name()

{

return this.first\_name;

}

public string GetId\_num()

{

return this.id\_num;

}

}

class Id\_Kid

{

private string id\_num;//מספר תעודת זהות

private string first\_name;//שם פרטי

private string last\_name;//שם משפחה

private Date birth;//תאריך לידה

private string country\_of\_birth;//ארץ לידה

private string nation;//לאום

public Id\_Kid (string id\_num, string first\_name, string last\_name, Date birth, string country\_of\_birth, string nation)

{

this.id\_num = id\_num;

this.first\_name = first\_name;

this.last\_name = last\_name;

this.birth = birth;

this.country\_of\_birth = country\_of\_birth;

this.nation = nation;

}

public string GetFirst\_name()

{

return this.first\_name;

}

public string GetId\_num()

{

return this.id\_num;

}

}

static void Main(string[] args)

{

/\*//p175 ex8

Id\_Kid ik = new Id\_Kid("2561", "saaey", "sd", new Date(1, 5, 1990), "r", "d");

Id\_Adult ia = new Id\_Adult("2561", "ey", "sd", new Date(1, 5, 1900),"t",2,new Id\_Kid(ik);

Console.WriteLine(ia.Add\_kid(ik));

ia.Eightiiln(ik);

ia.Add\_kid(ik);

Console.WriteLine(ia.Adults(ia)); \*/

Id\_Kid[] kid = new Id\_Kid[5];

Id\_Adult[] ad = new Id\_Adult[3];

for (int i = 0; i < ad.Length; i++)

{

for (int j = 0; j < kid.Length; j++)

{

Console.WriteLine(" string id\_num, string first\_name, string last\_name, Date birth, string country\_of\_birth, string nation");

kid[j]= new Id\_Kid(Console.ReadLine(), Console.ReadLine(), Console.ReadLine(),

new Date(int.Parse(Console.ReadLine()), int.Parse(Console.ReadLine()), int.Parse(Console.ReadLine())),

Console.ReadLine(), Console.ReadLine());

}

Console.WriteLine("(string id\_num, string first\_name, string last\_name, Date birth, string country\_of\_birth,int num\_kids)");

ad[i] = new Id\_Adult(Console.ReadLine(), Console.ReadLine(), Console.ReadLine(),

new Date(int.Parse(Console.ReadLine()), int.Parse(Console.ReadLine()), int.Parse(Console.ReadLine())),

Console.ReadLine(),3,kid);

}

for (int i = 0; i < ad.Length; i++)

{

for (int j = i; j < ad.Length; j++)

{

if (i!=j&&ad[i].Adults(ad[j])>0)

{

Console.WriteLine("1 "+ad[i].GetFirst\_name()+" "+ad[i].GetId\_num()+"2 "+ ad[j].GetFirst\_name() + " " + ad[j].GetId\_num());

}

}

}

}

}