

## RGB színek

Fényképek számítógépes tárolására sokféle képformátumot ismerünk. A legegyszerűbb típusok az eredeti kép pixeljeihez tartozó színértékeket tárolják, például ilyen a BMP vagy a fényképezőgépek által használt RAW formátum. Az ilyen képállomány bináris, melynek első néhány bájtja a kép jellemzőit írja le, míg a további bájtok sorfolytonosan a kép egy-egy pixelének színét adják meg.

Ebben a feladatban egy RAW formátumú fájlból átalakított szöveges állományt kell feldolgoznia. A *kep.txt* szöveges állomány egy  $640 \times 360$  méretű, RGB kódolású képet ír le. Az állomány csak a képpontok színét tartalmazza sorfolytonosan, azaz a fájl 360 sorának mindegyike 640 képpontból, képpontonként három színértékből áll. Az első szám a piros (**R**ed), a második szám a zöld (**G**reen) és a harmadik szám a kék (**B**lue) értéket tartalmazza. Az RGB értékek 0 és 255 közötti egészek, melyeket a fájlban egy-egy szóköz választ el.

Részlet az állomány első néhány sorából (az alábbi példákban szereplő számhármassok félkövér betűstílussal vannak kiemelve):

```
0 85 112 0 86 113 0 86 113 0 86 113 0 86 113 0 86 113 ... 0 86 111
0 86 111 0 86 113 0 86 113 0 86 113 0 86 113 ... 0 86 113 0 86 113
0 86 113 0 86 113 0 86 113 1 87 114 ... 0 86 113 0 86 113 0 86 113
0 86 113 1 87 114 1 87 114 ... ..
```

Az első sor első három száma a kép bal felső képpontjának színe, azaz RGB(0, 85, 112), míg a második sor harmadik pixelének színe RGB(0, 86, 113), a negyedik sor hatodik oszlopában lévő képpont színe RGB(1, 87, 114). A szöveges állomány által leírt kép:



Készítsen programot, amely az állomány adatait felhasználva megoldja az alábbi feladatokat! A program forráskódját mentse *rgb* néven! A program megírásakor a felhasználó által megadott adatok helyességét, érvényességét nem kell ellenőriznie, és feltételezheti, hogy a beolvasandó adatok a leírtaknak megfelelnek.

A képernyőre írást igénylő részfeladatok esetén – a mintához tartalmában hasonlóan – írja ki a képernyőre a feladat sorszámát (például: 2. feladat:), és utaljon a kiírt tartalomra is! Ha a felhasználótól kér be adatot, jelenítse meg a képernyőn, hogy milyen értéket vár! Mindkét esetben az ékezetmentes kiírás is elfogadott.

1. Olvassa be a *kep.txt* állomány tartalmát, és tárolja el a  $640 \times 360$  képpont színét!
2. Kérje be a felhasználótól a kép egy pontjának sor- és oszlopszámát (a számozás mindkét esetben 1-től indul), és írja a képernyőre az adott képpont RGB színösszetevőit a minta szerint!
3. Világosnak tekintjük az olyan képpontot, amely RGB-értékeinek összege 600-nál nagyobb. Számolja meg és írja ki, hogy a teljes képen hány világos képpont van!
4. A kép legsötétebb pontjainak azokat a pontokat tekintjük, amelyek RGB-értékeinek összege a legkisebb. Adja meg, hogy mennyi a legkisebb összeg, illetve keresse meg az ilyen RGB összegű pixeleket, és írja ki mindegyik színét RGB(r,g,b) formában a mintának megfelelően!
5. A képen a kék ég látható közepén egy felhővel. Az ég és a felhő színe között jelentős különbség van, így az ég-felhő határvonal programmal is felismerhető. Ennek megtalálásához készítsen függvényt *hatar* néven, amely megadja, hogy egy adott sorban van-e olyan hely a képen, ahol az egymás

melletti képpontok kék színösszetevőinek eltérése meghalad egy adott értéket! A függvény kapja meg paraméterként a sor számát, illetve az eltérés értékét, melyek egészek! A függvény visszatérési értéke egy logikai érték legyen, amely megadja, hogy az adott sorban volt-e az eltérést meghaladó különbség az egymás melletti képpontok kék színében!

- Keresse meg a képen a felhő első és utolsó sorát az előzőleg elkészített függvény segítségével úgy, hogy eltérésként 10-et ad meg a függvénynek bemenetként! Adja meg az első és az utolsó olyan sor sorszámát, ahol az eltérés a soron belül valahol 10-nél nagyobb!

Példa a szöveges kimenetek kialakításához:

```
2. feladat
Kérem egy képpont adatait!
Sor (1-360 között): 180
Oszlop (1-640 között): 320
A képpont színe RGB(184,183,181)
3. feladat:
A világos képpontok száma:7837
4. feladat:
A legsötétebb pont RGB összege: 197
A legsötétebb pixelek színe:
RGB(0,85,112)
RGB(0,86,111)
RGB(0,86,111)
6. feladat
A felhő legfelső sora: 103
A felhő legalsó sora: 280
```

Megoldás:

```
using System;
using System.IO;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Reflection;
namespace rgb
{
    internal class Program
    {
        public static string[,] rgbkodok = new string[360, 640]; //többszörös tömb
        /*A kep.txt szöveges állomány egy 640x360 méretű, RGB kódolású képet ír le. a fájl 360 sorának mindegyike 640 képpontból, képpontonként
        három színértékből áll.*/
        static void Main(string[] args)
        {
            int k = 0;
            //Olvassa be a kep.txt állomány tartalmát, és tárolja el a 640x360 képpont színét!
            string[] fajlbol = File.ReadAllLines("kep.txt", Encoding.UTF8); //beolvassa a kep.txt tartalmát
            for (int i = 0; i < fajlbol.Length; i++)
            {
                string[] sor = fajlbol[i].Split();
                k = 0;
                for (int j = 0; j < sor.Length; j += 3)
                {
                    rgbkodok[i, k] = sor[j] + "," + sor[j + 1] + "," + sor[j + 2];
                    //Az első szám a piros (Red), a második szám a zöld(Green) és a harmadik szám a kék(Blue) értéket tartalmazza.
                    k++;
                }
            }
            //Console.WriteLine(rgbkodok[0, 0]); //első kód
            //Console.WriteLine(rgbkodok[359, 639]); //utolsó kód
            // 2. feladat /*Kérje be a felhasználótól a kép egy pontjának sor- és oszlopszámát (a számozás mindkét
            esetben 1 - től indul), és írja a képernyőre az adott képpont RGB színösszetevőit a minta szerint!*/
            int sori = 0;
            int oszlopi = 0;
            Console.WriteLine("2. feladat\nKérem egy képpont adatait! ");
            do
            {
                Console.Write("Sor (1-360 között): ");
                sori = int.Parse(Console.ReadLine());
            }
            while (sori < 1 || sori > 360);
            do
            {
                Console.Write("Oszlop (1-640 között): ");
                oszlopi = int.Parse(Console.ReadLine());
            }
            while (oszlopi < 1 || oszlopi > 640);
        }
    }
}
```

```

Console.WriteLine("A képpont színe RGB(" + rgbkodok[sori - 1, oszlopi - 1] + ")");
/*Világosnak tekintjük az olyan képpontot, amely RGB-értékeinek összege 600-nál nagyobb.
Számolja meg és írja ki, hogy a teljes képen hány világos képpont van! */
//Console.WriteLine(rgbkodok[0, 0]);//első kód
int világoskeppontokszama = 0;//megszámlálás tétele
int osszeg = 0;
for (int i = 0; i < 360; i++)
{
    for (int j = 0; j < 640; j++)
    {
        osszeg = rgbosszeg(rgbkodok[i, j]);//a függvény meghatározza az RGB kódok összegét
        if (osszeg > 600) világoskeppontokszama++;
    }
}
Console.WriteLine("3. feladat:\nA világos képpontok száma:" + világoskeppontokszama);
/* A kép legsötétebb pontjainak azokat a pontokat tekintjük, amelyek RGB-értékeinek összege
a legkisebb. Adja meg, hogy mennyi a legkisebb összeg, illetve keresse meg az ilyen RGB
összegű pixeleket, és írja ki mindegyik színét RGB(r,g,b) formában a mintának
megfelelően! */
int legkisebbszeg = 1000;//minimumkiválasztás tétele
for (int i = 0; i < 360; i++)
{
    for (int j = 0; j < 640; j++)
    {
        osszeg = rgbosszeg(rgbkodok[i, j]);
        if (osszeg < legkisebbszeg) legkisebbszeg = osszeg;
    }
}
Console.WriteLine("4. feladat:\nA legsötétebb pont RGB összege: " + legkisebbszeg);
Console.WriteLine("A legsötétebb pixelek színe:");//kiválogatás tétele
for (int i = 0; i < 360; i++)
{
    for (int j = 0; j < 640; j++)
    {
        osszeg = rgbosszeg(rgbkodok[i, j]);
        if (osszeg == legkisebbszeg) Console.WriteLine("RGB(" + rgbkodok[i, j] + ")");
    }
}
/*A képen a kék ég látható közepén egy felhővel. Az ég és a felhő színe között jelentős
különbség van, így az ég-felhő határvonal programmal is felismerhető. Ennek
megtalálásához készítsen függvényt hatar néven, amely megadja, hogy egy adott sorban
van-e olyan hely a képen, ahol az egymás melletti képpontok kék színösszetevőinek eltérése
meghalad egy adott értéket! A függvény kapja meg paraméterként a sor számát, illetve
az eltérés értékét, melyek egészek! A függvény visszatérési értéke egy logikai érték legyen,
amely megadja, hogy az adott sorban volt-e az eltérést meghaladó különbség az egymás
melletti képpontok kék színében! */
int felso = 1, also = 360;
while (!hatar(felso, 10)) felso++;
while (!hatar(also, 10)) also--;
Console.WriteLine("6. feladat\nA felhő legfelső sora: " + felso);
Console.WriteLine("A felhő legalsó sora: " + also);
Console.ReadKey();
}
private static int rgbosszeg(string rgbkodja)//a függvény meghatározza az RGB kódok összegét
{
    int o = 0;//összeg lesz a visszatérési érték
    string[] rgbelemek = rgbkodja.Split(',');//az RGB kódok , -vel vannak elválasztva
    for (int k = 0; k < rgbelemek.Length; k++)
    {
        o += Convert.ToInt32(rgbelemek[k]);
    }
    return o;
}
private static bool hatar(int sorszam, int elteres)// A függvény kapja meg paraméterként a sor számát, illetve az eltérés értékét, melyek
egészek
{
    for (int i = 0; i < 640; i += 2)
    {
        string[] rgbelemek = rgbkodok[sorszam - 1, i].Split(',');
        string[] rgbelemek2 = rgbkodok[sorszam - 1, i + 1].Split(',');
        int kulonbseg = Math.Abs(Convert.ToInt32(rgbelemek[2]) - Convert.ToInt32(rgbelemek2[2]));
        if (kulonbseg > elteres) return true;
        //A függvény visszatérési értéke egy logikai érték legyen,
        //amely megadja, hogy az adott sorban volt-e az eltérést meghaladó különbség az egymásmelletti képpontok kék színében!
    }
    return false;
}
}
}
}

```