**Tartalom**

[Összes C# kódom 4](#_Toc134903368)

[Command-line interface (CLI) vagyis konzolos progik 4](#_Toc134903369)

[001. neveskoszones - Név bekérése és köszönés 4](#_Toc134903370)

[002. alakzatok1 - Tíz darab \*-ot és tíz darab o-t ír ki. 4](#_Toc134903371)

[003. alakzatok2 - Öt darab \*o két sorban 5](#_Toc134903372)

[004. ketszamosszege - Hát két szám összege :D 5](#_Toc134903373)

[005. haromszamosszege - Hát három szám összege :D 6](#_Toc134903374)

[006. ketszamkulonbsege - Két szám különbsége 6](#_Toc134903375)

[007. ketszamszorzata - Két szám szorzata 7](#_Toc134903376)

[008. hanyadosplusztizedes - Egész számok és tizedes törtek osztására 7](#_Toc134903377)

[009. nullatolezerig - 0-tól 1000-ig kiírja a számokat 8](#_Toc134903378)

[010. szaztolnyolcszazig - 100-tól 800-ig írja ki a számokat 8](#_Toc134903379)

[011. elsotiztermszam - Első 10 természetes szám 9](#_Toc134903380)

[012. csakparos - 10-ig a páros számok 9](#_Toc134903381)

[013. harommaloszthatoszazig - 100-ig a 3-mal osztható számok 10](#_Toc134903382)

[014. dupla - Bekér egy számot a felhasználótól és megduplázza azt 10](#_Toc134903383)

[015. elsootszamosszege - Az első 5 szám összege 11](#_Toc134903384)

[016. szamoknegyzeteegytoltizig - 1-től 10-ig a számok négyzete 11](#_Toc134903385)

[017. szamoknegyzeteezerig - 1000-ig minden szám négyzete 12](#_Toc134903386)

[017.\_2. negyzetezerigwhileciklussal - 1000-ig minden szám négyzete while ciklussal 12](#_Toc134903387)

[018. parosegytolszazig - 1-től 100-ig a páros számok 13](#_Toc134903388)

[019. ketszamnegyzete - Bekér két számot és kiírja a négyzetüket 13](#_Toc134903389)

[020. parosotventoltizig - Páros számok 50-től 10-ig 14](#_Toc134903390)

[021. negyzetkerter - A négyzet kerülete és területe 14](#_Toc134903391)

[022. teglalapkerter - A téglalap kerülete és területe 15](#_Toc134903392)

[023. elsotizparososszege - Az első 10 páros szám összege 15](#_Toc134903393)

[024. paratlannyolcvantolhuszig - 80-tól 20-ig a páratlan számok 16](#_Toc134903394)

[025. atlag - 3 szám átlaga 16](#_Toc134903395)

[026. paratlanszamokegytolkilencvenig - 1-től 90-ig a páratlan számok 17](#_Toc134903396)

[027. paratlanszamokhatvantoltizenotig - 60-tól 15-ig a páratlan számok 17](#_Toc134903397)

[028. parosszamokhetventolhuszonotig - 70-től 25-ig a páros számok 18](#_Toc134903398)

[029. eddigird - Felhasználó beír egy számot, a progi meg addig a számig kiírja 18](#_Toc134903399)

[030. kettoszamkozulanagyobbik - Két szám közül a nagyobbikat írja ki 19](#_Toc134903400)

[031. haromkozulalegkisebb - Három szám közül a legkisebb 19](#_Toc134903401)

[032. szerkeszhetoeharomszog - Eldönti, hogy szerkeszthető-e △ 20](#_Toc134903402)

[033. masodfokuegyenletyoutubeos - Másodfokú egyenlet (CLI) 21](#_Toc134903403)

[033.\_2. masodfokuegyenletyoutubeosgrafikus - Másodfokú egyenlet (GUI) 22](#_Toc134903404)

[Form1.cs 22](#_Toc134903405)

[A program futás közben: 24](#_Toc134903406)

[034. haromszogatfogojanakmeghatarozasa - △ átfogójának meghatározása 25](#_Toc134903407)

[035. reciprok - Számok reciprok értéke 25](#_Toc134903408)

[036. hatvanyertek - Hatványérték kiszámolása 26](#_Toc134903409)

[037. csakpozitiv - Csak pozitív számokat hajlandó kiírni 26](#_Toc134903410)

[038. ketszamkoztitavolsag - Két szám közti távolság 27](#_Toc134903411)

[039. ketszamatlaga - Két szám átlaga 27](#_Toc134903412)

[040. migazosszegnemszaz - Két szám összegétől számol felfelé 100-ig 28](#_Toc134903413)

[042. randomszamok - 0-tól 100-ig véletlen számokat generál 28](#_Toc134903414)

[043. véletlenszamoktiztolotvenig - 10-től 50-ig véletlen számokat generál 29](#_Toc134903415)

[044. veletlenszamokszazharminckettotolszaznegyvenhetig - 132-től 147-ig véletlen számokat generál 30](#_Toc134903416)

[045. veletlen132148csakparos - 132-től 148-ig csak páros véletlen számok 30](#_Toc134903417)

[046. dobokocka 31](#_Toc134903418)

[047. primszame - Eldönti, hogy a kapott szám prímszám-e 33](#_Toc134903419)

[048. rovidebbszo - Két szó közül eldönti, hogy melyik a rövidebb 34](#_Toc134903420)

[049. azonosakeaszavak - Két szó közül eldönti, hogy azonosak-e 35](#_Toc134903421)

[050. ahosszabbikszo - Két szó közül a hosszabbikat írja ki 36](#_Toc134903422)

[051. ketszoazonoskarakterei - Két szó azonos karakterei (hiányos!) 37](#_Toc134903423)

[059. TombosFeladat - Tömbös feladat - Sebesség 38](#_Toc134903424)

[065. Fajlkezeles - Fájlkezelés 39](#_Toc134903425)

[066. gamerek - Játékosok 40](#_Toc134903426)

[067. gamerek2 - Másik gamer osztály - Négy játékos és pontjaik 41](#_Toc134903427)

[Class1.cs 41](#_Toc134903428)

[Program.cs 41](#_Toc134903429)

[068. emberosztaly - 4 ember neve, életkora, a legidősebb és a legfiatalabb neve és életkora 42](#_Toc134903430)

[069. helsinki1952 - Olimpia, érmek, helyezések stb. 44](#_Toc134903431)

[eredmeny.cs 44](#_Toc134903432)

[Program.cs 44](#_Toc134903433)

[071 domino - Dominó 47](#_Toc134903434)

[072. sudokuCLI - Sudoku konzolos 49](#_Toc134903435)

[Osztaly.cs 49](#_Toc134903436)

[Program.cs 49](#_Toc134903437)

[074. IskolaGUI - Iskola tanulóinak listája és adataik 52](#_Toc134903438)

[Form1.cs 52](#_Toc134903439)

[Program.cs 54](#_Toc134903440)

[075. fizetes - Elégedett vagy-e a fizetéseddel? 54](#_Toc134903441)

[Form1.cs 54](#_Toc134903442)

[Program.cs 55](#_Toc134903443)

[A program futás közben: 56](#_Toc134903444)

[076. Paint alapok 57](#_Toc134903445)

[Form1.cs 57](#_Toc134903446)

[A program futás közben: 59](#_Toc134903447)

[084. godrok - Gödrök adatai (CLI-s, konzolos feladat) 60](#_Toc134903448)

[Program.cs 60](#_Toc134903449)

[092. Tesztverseny - Konzolos 64](#_Toc134903450)

[Program.cs 64](#_Toc134903451)

# Összes C# kódom

# Command-line interface (CLI) vagyis konzolos progik

# 001. neveskoszones - Név bekérése és köszönés

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_001.neveskoszones

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Kérem a neved és köszönök neked!");

string nev = Console.ReadLine();

Console.WriteLine("Hali, csá, csövi " + nev + "!");

Console.ReadLine();

}

}

}

# 002. alakzatok1 - Tíz darab \*-ot és tíz darab o-t ír ki.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_002.alakzatok1

{

class Program

{

static void Main(string[] args)

{

for (int i = 0; i < 11; i++)

{

Console.Write("\*");

}

Console.WriteLine("\n");

for (int i = 0; i < 11; i++)

{

Console.Write("o");

}

Console.ReadKey();

}

}

}

# 003. alakzatok2 - Öt darab \*o két sorban

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_003.alakzatok2

{

class Program

{

static void Main(string[] args)

{

for (int i = 0; i < 5; i++)

{

Console.Write("\*o");

}

Console.WriteLine("\n");

for (int i = 0; i < 5; i++)

{

Console.Write("\*o");

}

Console.ReadKey();

}

}

}

# 004. ketszamosszege - Hát két szám összege :D

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_004.ketszamosszege

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Kérek egy számot!");

int szam1 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Kérek még egy számot!");

int szam2 = Convert.ToInt32(Console.ReadLine());

int osszeg = szam1 + szam2;

Console.WriteLine("A két szám összege: " + osszeg + "." + "\n");

Console.ReadKey();

}

}

}

# 005. haromszamosszege - Hát három szám összege :D

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_004.ketszamosszege

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Kérek egy számot!");

int szam1 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Kérek még egy számot!");

int szam2 = Convert.ToInt32(Console.ReadLine());

int osszeg = szam1 + szam2;

Console.WriteLine("A két szám összege: " + osszeg + "." + "\n");

Console.ReadKey();

}

}

}

# 006. ketszamkulonbsege - Két szám különbsége

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_006.ketszamkulonbsege

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Kérek egy számot!");

int szam1 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Kérek még egy számot!");

int szam2 = Convert.ToInt32(Console.ReadLine());

int kivonas = szam1 - szam2;

Console.WriteLine("A két szám különbsége: " +kivonas + "." + "\n");

Console.ReadKey();

}

}

}

# 007. ketszamszorzata - Két szám szorzata

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_007.ketszamszorzata

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Kérek egy számot! ");

int szam = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Kérek még egy számot! ");

int szam2 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("A két szám szorzata: " + szam \* szam2 + ".");

Console.ReadKey();

}

}

}

# 008. hanyadosplusztizedes - Egész számok és tizedes törtek osztására

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_008.hanyadosplusztizedes

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Kérek egy számot!");

double szam1 = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("Kérek még egy számot!");

double szam2 = Convert.ToDouble(Console.ReadLine());

double osztas = szam1 / szam2;

Console.WriteLine("A két szám hányadosa: " + osztas + "." + "\n");

Console.ReadKey();

}

}

}

# 009. nullatolezerig - 0-tól 1000-ig kiírja a számokat

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_009.nullatolezerig

{

class Program

{

static void Main(string[] args)

{

for (int i = 0; i < 1001; i++)

{

Console.WriteLine(i);

}

Console.ReadKey();

}

}

}

# 010. szaztolnyolcszazig - 100-tól 800-ig írja ki a számokat

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_010.szaztolnyolcszazig

{

class Program

{

static void Main(string[] args)

{

for (int i = 100; i < 801; i++)

{

Console.WriteLine(i);

}

Console.ReadKey();

}

}

}

# 011. elsotiztermszam - Első 10 természetes szám

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_011.elsotiztermszam

{

class Program

{

static void Main(string[] args)

{

for (int i = 0; i < 11; i++)

{

Console.WriteLine(i);

}

Console.ReadKey();

}

}

}

# 012. csakparos - 10-ig a páros számok

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_012.csakparos

{

class Program

{

static void Main(string[] args)

{

for (int i = 1; i < 11; i++)

if (i % 2 == 0)

{

Console.WriteLine(i);

}

Console.ReadKey();

}

}

}

# 013. harommaloszthatoszazig - 100-ig a 3-mal osztható számok

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_013.harommaloszthatoszazig

{

class Program

{

static void Main(string[] args)

{

for (int i = 1; i < 101; i++)

if (i % 3 == 0)

{

Console.WriteLine(i);

}

Console.ReadKey();

}

}

}

# 014. dupla - Bekér egy számot a felhasználótól és megduplázza azt

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_014.dupla

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Kérek egy számot és kiírom a dupláját! ");

int szam = Convert.ToInt32(Console.ReadLine());

Console.WriteLine(2 \* szam);

Console.ReadKey();

}

}

}

# 015. elsootszamosszege - Az első 5 szám összege

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_015.elsootszamosszege

{

class Program

{

static void Main(string[] args)

{

int osszeadelsoot = 0;

for (int i = 0; i < 6; i++)

{

osszeadelsoot = osszeadelsoot + i;

}

Console.WriteLine("Az első öt szám összege: " + osszeadelsoot + ".");

Console.ReadKey();

}

}

}

# 016. szamoknegyzeteegytoltizig - 1-től 10-ig a számok négyzete

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_016.szamoknegyzeteegytoltizig

{

class Program

{

static void Main(string[] args)

{

for (int i = 0; i < 11; i++)

{

Console.WriteLine("A(z) " + i + " négyzete: " + i \* i);

}

Console.ReadKey();

}

}

}

# 017. szamoknegyzeteezerig - 1000-ig minden szám négyzete

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_017.szamoknegyzeteezerig

{

class Program

{

static void Main(string[] args)

{

for (int i = 0; i < 1001; i++)

{

Console.WriteLine("A(z) " + i + " négyzete: " + i \* i);

}

Console.ReadKey();

}

}

}

# 017.\_2. negyzetezerigwhileciklussal - 1000-ig minden szám négyzete while ciklussal

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_017.\_2.negyzetezerigwhileciklussal

{

class Program

{

static void Main(string[] args)

{

int i = 1;

while (i \* i < 1000)

{

Console.WriteLine("A(z) " + i + " négyzete: " + i \* i);

i++;

}

Console.ReadKey();

}

}

}

# 018. parosegytolszazig - 1-től 100-ig a páros számok

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_018.parosegytolszazig

{

class Program

{

static void Main(string[] args)

{

for (int i = 1; i < 101; i++)

if (i % 2 == 0)

{

Console.WriteLine(i);

}

Console.ReadKey();

}

}

}

# 019. ketszamnegyzete - Bekér két számot és kiírja a négyzetüket

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_019.ketszamnegyzete

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Kérek egy számot és kiírom a négyzetét!");

double szam1 = Convert.ToDouble(Console.ReadLine());

double negyzet = szam1 \* szam1;

Console.WriteLine("A szám négyzete: " +negyzet + "." + "\n");

Console.WriteLine("Kérek még egy számot és kiírom ennek is a négyzetét!");

double szam2 = Convert.ToDouble(Console.ReadLine());

double negyzet2 = szam2 \* szam2;

Console.WriteLine("A szám négyzete: " +negyzet2 + "." + "\n");

Console.ReadKey();

}

}

}

# 020. parosotventoltizig - Páros számok 50-től 10-ig

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_020.parosotventoltizig

{

class Program

{

static void Main(string[] args)

{

for (int i = 50; i > 9; i--)

if (i % 2 == 0)

{

Console.WriteLine(i);

}

Console.ReadKey();

}

}

}

# 021. negyzetkerter - A négyzet kerülete és területe

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_021.negyzetkerter

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Kérem a négyzet oldalát és kiírom annak kerületét!");

int szam1 = Convert.ToInt32(Console.ReadLine());

int kerulet = szam1 \* 4;

Console.WriteLine("A négyzet kerülete " + kerulet + " cm." + "\n");

Console.WriteLine("Kérem a négyzet oldalát és kiírom annak területét!");

int szam2 = Convert.ToInt32(Console.ReadLine());

int terulet = szam2 \* szam2;

Console.WriteLine("A négyzet területe " + terulet + " cm2." + "\n");

Console.ReadKey();

}

}

}

# 022. teglalapkerter - A téglalap kerülete és területe

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_022.teglalapkerter

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Kérem a téglalap egyik oldalát!");

int a = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Kérem a téglalap másik oldalát és kiírom a területét!");

int b = Convert.ToInt32(Console.ReadLine());

int terulet = a \* b;

Console.WriteLine("A négyzet kerülete " + terulet + " cm2." + "\n");

Console.WriteLine("Kérem a téglalap egyik oldalát!");

int a2 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Kérem a téglalap másik oldalát és kiírom a kerületét!");

int b2 = Convert.ToInt32(Console.ReadLine());

int kerulet = (a2 + b2) \* 2;

Console.WriteLine("A négyzet területe " + kerulet + " cm." + "\n");

Console.ReadKey();

}

}

}

# 023. elsotizparososszege - Az első 10 páros szám összege

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_023.elsotizparososszege

{

class Program

{

static void Main(string[] args)

{

int szamlalo = 0;

int osszeg = 0;

for (int i = 0; i < 101; i++)

{

if (i % 2 == 0)

{

szamlalo = szamlalo + 1;

osszeg = osszeg + i;

if (szamlalo > 10)

{

break;

}

}

}

Console.WriteLine(osszeg);

Console.ReadKey();

}

}

}

# 024. paratlannyolcvantolhuszig - 80-tól 20-ig a páratlan számok

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_024.paratlannyolcvantolhuszig

{

class Program

{

static void Main(string[] args)

{

for (int i = 80; i > 19; i--)

if (i % 2 != 0)

{

Console.WriteLine(i);

}

Console.ReadKey();

}

}

}

# 025. atlag - 3 szám átlaga

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace atlag

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Kérek egy számot!");

int szam1 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Kérek még egy számot!");

int szam2 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Kérek egy harmadik számot!");

int szam3 = Convert.ToInt32(Console.ReadLine());

int atlag = (szam1 + szam2 + szam3) / 3;

Console.WriteLine("A három szám átlaga:" +atlag + "\n");

Console.ReadKey();

}

}

}

# 026. paratlanszamokegytolkilencvenig - 1-től 90-ig a páratlan számok

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_026.paratlanszamokegytolkilencvenig

{

class Program

{

static void Main(string[] args)

{

for (int i = 0; i < 91; i++)

{

if (i % 2 != 0)

Console.WriteLine(i);

}

Console.ReadKey();

}

}

}

# 027. paratlanszamokhatvantoltizenotig - 60-tól 15-ig a páratlan számok

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_027.paratlanszamokhatvantoltizenotig

{

class Program

{

static void Main(string[] args)

{

for (int i = 60; i > 14; i--)

{

if (i % 2 != 0)

Console.WriteLine(i);

}

Console.ReadKey();

}

}

}

# 028. parosszamokhetventolhuszonotig - 70-től 25-ig a páros számok

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_028.parosszamokhetventolhuszonotig

{

class Program

{

static void Main(string[] args)

{

for (int i = 70; i > 24; i--)

{

if (i % 2 == 0)

Console.WriteLine(i);

}

Console.ReadKey();

}

}

}

# 029. eddigird - Felhasználó beír egy számot, a progi meg addig a számig kiírja

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_029.eddigird

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Kérek egy számot! ");

int szam = Convert.ToInt32(Console.ReadLine());

Console.Write("\n");

int eddigird = 1;

while (eddigird < szam)

//for (int i = 0; i < szam; i++)

{

Console.WriteLine(eddigird);

eddigird++;

}

Console.ReadKey();

}

}

}

# 030. kettoszamkozulanagyobbik - Két szám közül a nagyobbikat írja ki

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_030.kettoszamkozulanagyobbik

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Kérek egy számot!");

int szam1 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Kérek még egy számot!");

int szam2 = Convert.ToInt32(Console.ReadLine());

if (szam1 > szam2)

Console.WriteLine("A nagyobbik szám: " + szam1 + "." + "\n");

else

Console.WriteLine("A nagyobbik szám: " + szam2 + "." + "\n");

Console.ReadKey();

}

}

}

# 031. haromkozulalegkisebb - Három szám közül a legkisebb

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_031.haromkozulalegkisebb

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Kérek egy számot!");

int szam1 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Kérek még egy számot!");

int szam2 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Kérek egy harmadik számot is!");

int szam3 = Convert.ToInt32(Console.ReadLine());

{

if (szam1 < szam2 && szam1 < szam3)

Console.WriteLine("A legkisebb szám: " + szam1 + "." + "\n");

else if (szam2 < szam3 && szam2 < szam1)

Console.WriteLine("A legkisebb szám: " + szam2 + "." + "\n");

else

Console.WriteLine("A legkisebb szám: " + szam3 + "." + "\n");

}

Console.ReadKey();

}

}

}

# 032. szerkeszhetoeharomszog - Eldönti, hogy szerkeszthető-e △

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_032.szerkeszhetoeharomszog

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Kérem a háromszög egyik befogóját! ");

int befogo1 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Kérem a háromszög másik befogóját! ");

int befogo2 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Kérem a háromszög átfogóját! ");

int atfogo = Convert.ToInt32(Console.ReadLine());

{

if (befogo1 + befogo2 > atfogo && befogo1 + atfogo > befogo2)

Console.WriteLine("Ezekkel az adatokkal szerkeszthető háromszög.");

else if (befogo2 + atfogo > befogo1 && befogo2 + befogo1 > atfogo && atfogo + befogo2 > befogo1)

Console.WriteLine("Ezekkel az adatokkal szerkeszthető háromszög.");

else

Console.WriteLine("Ezekkel az adatokkal nem szerkeszthető háromszög.");

}

Console.ReadKey();

}

}

}

# 033. masodfokuegyenletyoutubeos - Másodfokú egyenlet (CLI)

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_033.masodfokuegyenletyoutubeos

{

class Program

{

static void Main(string[] args)

{

double a, b, c, d, x1, x2;

Console.Write("Add meg az a értékét! ");

a = Convert.ToDouble(Console.ReadLine());

if (a == 0)

{

Console.WriteLine("\n Ez az egyenlet így nem lehet másodfokú!");

}

else

{

Console.Write("\n Add meg a b értékét! ");

b = Convert.ToDouble(Console.ReadLine());

Console.Write("\n Add meg a c értékét! ");

c = Convert.ToDouble(Console.ReadLine());

d = b \* b - 4 \* a \* c;

if(d<0)

{

Console.WriteLine("Nincs valós megoldás!");

}

else

{

x1 = (-b + Math.Sqrt(d)) / (2 \* a);

x2 = (-b + Math.Sqrt(d)) / (2 \* a);

Console.WriteLine("X1={0}\tX2={1}", x1, x2);

}

}

Console.WriteLine("A program enter leütése után kilép!");

Console.ReadKey();

}

}

}

# 033.\_2. masodfokuegyenletyoutubeosgrafikus - Másodfokú egyenlet (GUI)

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace \_033.\_2.masodfokuegyenletyoutubeosgrafikus

{

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

# Form1.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace \_033.\_2.masodfokuegyenletyoutubeosgrafikus

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

try

{

double a, b, c, x1, x2;

a = double.Parse(textBox1.Text);

b = double.Parse(textBox2.Text);

c = double.Parse(textBox3.Text);

if (a == 0) throw new DivideByZeroException();

if (Math.Pow(b, 2) - 4 \* a \* c < 0)

{

throw new ArithmeticException("Nincs valós gyök.");

}

if (Math.Pow(b, 2) - 4 \* a \* c == 0)

{

x1 = -b / (2 \* a);

MessageBox.Show("A másodfokú egyenlet megoldása: " + x1);

}

else

{

x1 = (-b + Math.Sqrt(Math.Pow(b, 2) - 4 \* a \* c)) / (2 \* a);

x2 = (-b - Math.Sqrt(Math.Pow(b, 2) - 4 \* a \* c)) / (2 \* a);

MessageBox.Show("A másodfokú egyenlet megoldásai: x1=" + x1 + " x2=" + x2);

}

}

catch(FormatException)

{

MessageBox.Show("Hibás adatmegadás, vagy üres valamelyik szövegdoboz!");

}

catch (DivideByZeroException)

{

MessageBox.Show("Nem másodfokú az egyenlet!");

}

catch (ArithmeticException ar)

{

MessageBox.Show(ar.Message);

}

finally

{

this.ActiveControl = textBox1;

}

}

private void button2\_Click(object sender, EventArgs e)

{

textBox1.Text = textBox2.Text = textBox3.Text = "";

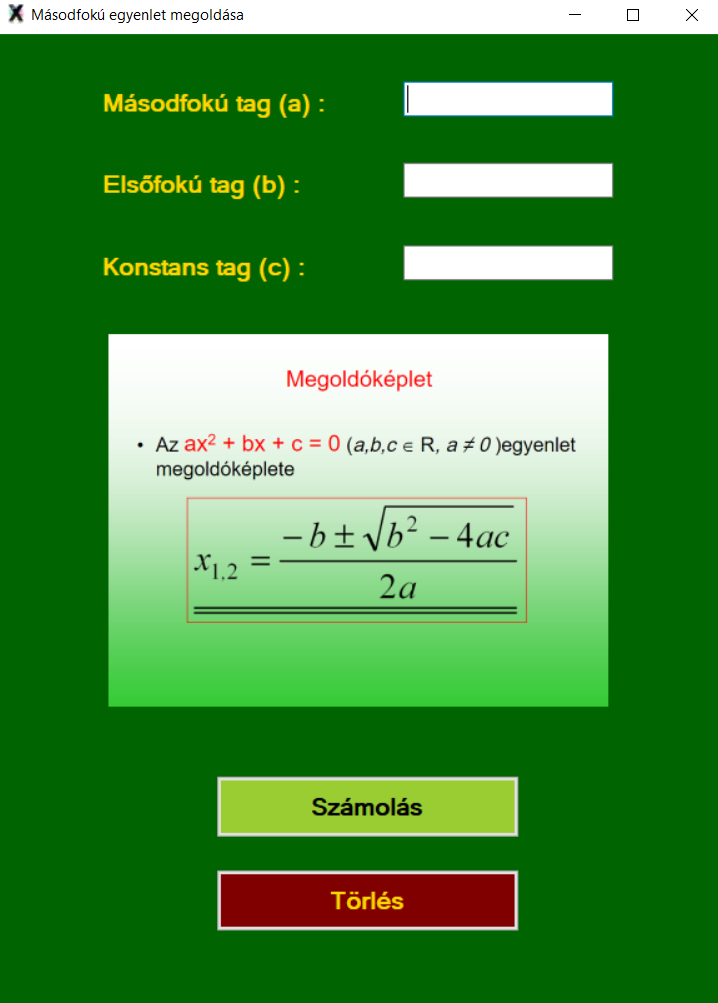
this.ActiveControl = textBox1;

}

}

}

# A program futás közben:



# 034. haromszogatfogojanakmeghatarozasa - △ átfogójának meghatározása

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_034.haromszogatfogojanakmeghatarozasa

{

class Program

{

static void Main(string[] args)

{

double a, b, c;

Console.WriteLine("Add meg a háromszög egyik befogójának értékét! ");

a = double.Parse(Console.ReadLine());

Console.WriteLine("Add meg a háromszög másik befogójának értékét! ");

b = double.Parse(Console.ReadLine());

c = Math.Sqrt(a \* a + b \* b);

c = Math.Round(c, 2);

Console.WriteLine("Az átfogó hossza:{0}", c);

Console.ReadKey();

}

}

}

# 035. reciprok - Számok reciprok értéke

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_035.reciprok

{

class Program

{

static void Main(string[] args)

{

double a;

Console.WriteLine("Kérek egy számot és kiírom a reciprokát!");

Console.Write("a=");

a = double.Parse(Console.ReadLine());

double c = 1 / a;

Console.WriteLine("A számod reciproka: " + c);

Console.ReadKey();

}

}

}

# 036. hatvanyertek - Hatványérték kiszámolása

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_036.hatvanyertek

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Kérem a hatvány alapot! ");

int alap = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Kérem a hatvány kitevőt! ");

int kitevo = Convert.ToInt32(Console.ReadLine());

double pow\_ab = Math.Pow(alap, kitevo);

Console.WriteLine("A hatványérték: " + pow\_ab);

Console.ReadLine();

}

}

}

# 037. csakpozitiv - Csak pozitív számokat hajlandó kiírni

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_037.csakpozitiv

{

class Program

{

static void Main(string[] args)

{

string strLine;

int intA;

do

{

Console.Clear();

Console.Write("Kérek egy pozitív számot, mert negatívakat nem vagyok hajlandó kiírni! ");

strLine = Console.ReadLine();

intA = Convert.ToInt32(strLine);

if (intA <= 0)

{

Console.WriteLine("Pozitív számot írj be!");

}

} while (intA > 0);

Console.ReadLine();

}

}

}

# 038. ketszamkoztitavolsag - Két szám közti távolság

using System;

namespace \_038.ketszamkoztitavolsag

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Kérek egy számot!");

int szam1 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Kérek még egy számot!");

int szam2 = Convert.ToInt32(Console.ReadLine());

double tavolsag = Math.Abs(szam1 - szam2);

Console.Write("A két szám közti távolság: " + tavolsag);

Console.ReadKey();

}

}

}

# 039. ketszamatlaga - Két szám átlaga

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_039.ketszamatlaga

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Kérek egy számot!");

int szam1 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Kérek még egy számot!");

int szam2 = Convert.ToInt32(Console.ReadLine());

int atlag = (szam1 + szam2) / 2;

Console.WriteLine("A számok átlaga: " + atlag);

Console.ReadKey();

}

}

}

# 040. migazosszegnemszaz - Két szám összegétől számol felfelé 100-ig

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_040.migazosszegnemszaz

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Kérek egy számot!");

int szam1 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Kérek még egy számot!");

int szam2 = Convert.ToInt32(Console.ReadLine());

int i = szam1 + szam2;

while (i < 101)

{

Console.WriteLine(i);

i++;

}

Console.ReadKey();

}

}

}

# 042. randomszamok - 0-tól 100-ig véletlen számokat generál

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_042.randomszamok

{

class Program

{

static void Main(string[] args)

{

Random random = new Random();

for (int i = 0; i < 100; i++)

{

Console.WriteLine(random.Next(0, 100));

}

//int num = random.Next(1, 7); // 1-től 7-ig egész számokat generál

//double num = random.NextDouble(); //tizedes törteket generál

//Console.WriteLine(num);

Console.ReadKey();

}

}

}

# 043. véletlenszamoktiztolotvenig - 10-től 50-ig véletlen számokat generál

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_043.véletlenszamoktiztolotvenig

{

class Program

{

static void Main(string[] args)

{

Random random = new Random();

for (int i = 10; i < 50; i++) // Ez a sor 40 db random számot fog kiírni, mert a 10 és az 50 közti intervallumot veszi.

{

Console.WriteLine(random.Next(10, 50)); // Ha a zárójelbe nem írunk paramétereket, akkor az integer (int) vagyis 2 milliárdig generálja a véletlen számokat, de a for ciklusban megadott paraméterek miatt akkor is csak 40 db számot fog kiírni. Viszont nekünk nem kellenek ekkora számok, mert a feladat azt mondja, hogy 10-től 50-ig generáljon számokat, így ide is be kell írni a kért paramétereket.

}

//int num = random.Next(1, 7); // 1-től 7-ig egész számokat generál

//double num = random.NextDouble(); //tizedes törteket generál

//Console.WriteLine(num);

Console.ReadKey();

}

}

}

# 044. veletlenszamokszazharminckettotolszaznegyvenhetig - 132-től 147-ig véletlen számokat generál

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_044.veletlenszamokszazharminckettotolszaznegyvenhetig

{

class Program

{

static void Main(string[] args)

{

{

Random random = new Random();

for (int i = 132; i < 148; i++) // Ez a sor 16 db random számot fog kiírni, mert a 132 és a 147 közti intervallumot veszi, ami 16.

{

Console.WriteLine(random.Next(132, 147)); // Ha a zárójelbe nem írunk paramétereket, akkor az integer (int) értékéig, vagyis 2 milliárdig generálja a véletlen számokat, de a for ciklusban megadott paraméterek miatt akkor is csak 16 db számot fog kiírni. Viszont nekünk nem kellenek ekkora számok, mert a feladat azt mondja, hogy 132-től 147-ig generáljon számokat, így ide is be kell írni a kért paramétereket.

}

//int num = random.Next(1, 7); // 1-től 7-ig egész számokat generál

//double num = random.NextDouble(); //tizedes törteket generál

//Console.WriteLine(num);

Console.ReadKey();

}

}

}

}

# 045. veletlen132148csakparos - 132-től 148-ig csak páros véletlen számok

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_045.veletlen132148csakparos

{

class Program

{

static void Main(string[] args)

{

Random random = new Random();

for (int i = 132; i < 149; i++) // 9 darab páros számot fog generálni véletlen sorrendben, mert a 132 és a 148 között 9 db páros szám van. De van amelyiket kétszer is legenerálja.

if (i % 2 == 0)

{

Console.WriteLine(random.Next(132, 149));

}

Console.ReadKey();

}

}

}

# 046. dobokocka

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_046.dobokocka

{

class Program

{

static void Main(string[] args)

{

#region Leírás

Console.WriteLine("Ez a program dobókocka dobálást szimulál és statisztikát készít belőle.");

Console.WriteLine("A folytatáshoz nyomj entert!");

Console.ReadLine();

Console.Clear();

#endregion Dobásszám bekérése

Console.Write("Add meg a dobások számát! ");

int dobasszam = Convert.ToInt32(Console.ReadLine());

//#endregion

#region Egyéb változók létrehozása

// Dobások száma adott értékre ezekben a változókban lesznek letárolva.

int egyesek\_szama = 0;

int kettesek\_szama = 0;

int harmasok\_szama = 0;

int negyesek\_szama = 0;

int otosok\_szama = 0;

int hatosok\_szama = 0;

#endregion

#region Dobások legenerálása, kiírása

Random randomszam = new Random();

// Létrehozom a tömböt, melyben letárolom az eredményeket.

int[] dobasok = new int[dobasszam];

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

{

dobasok[i] = randomszam.Next(1, 7);

Console.WriteLine("{0}. dobás:{1}", i+1, dobasok[i]);

System.Threading.Thread.Sleep(200);

}

Console.WriteLine("A statisztikához nyomj entert!");

Console.ReadLine();

Console.Clear();

#endregion

#region Statisztika készítése

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

{

if(dobasok[i] == 1)

{

egyesek\_szama++;

}

else if (dobasok[i] ==2)

{

kettesek\_szama++;

}

else if (dobasok[i] == 3)

{

harmasok\_szama++;

}

else if (dobasok[i] == 4)

{

negyesek\_szama++;

}

else if (dobasok[i] == 5)

{

otosok\_szama++;

}

else

{

//Console.WriteLine("Hatost dobtál, vége a játéknak.");

hatosok\_szama++;

}

}

#endregion

#region Statisztika kiíratása

Console.WriteLine("Statisztika:");

Console.WriteLine("Egyes: {0}\nKettes: {1}\nHármas: {2}\nNégyes: {3}\nÖtös: {4}\nHatos: {5}", egyesek\_szama, kettesek\_szama, harmasok\_szama, negyesek\_szama, otosok\_szama, hatosok\_szama);

#endregion

Console.ReadLine();

}

}

}

# 047. primszame - Eldönti, hogy a kapott szám prímszám-e

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_047.primszame

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Kérek egy számot!");

int szam = Convert.ToInt32(Console.ReadLine());

//Console.Clear();

int oszto = 0;

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

{

if (szam % i == 0)

{

oszto++;

}

else

{

}

}

if(oszto == 2)

{

Console.WriteLine("A szám prímszám mert 1-en és önmagán kívül más számokkal nem osztható.");

}

else

{

Console.WriteLine("A szám nem prímszám, mert 1-en és önmagán kívül más számokkal is osztható.");

}

Console.ReadKey();

}

}

}

# 048. rovidebbszo - Két szó közül eldönti, hogy melyik a rövidebb

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_048.rovidebbszo

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Kérek egy szót! ");

string szo1 = Convert.ToString(Console.ReadLine());

Console.WriteLine("Kérek még egy szót és kiírom a rövidebbet! ");

string szo2 = Convert.ToString(Console.ReadLine());

if(szo1.Length < szo2.Length)

{

Console.WriteLine("Az első szó " + szo1 + " rövidebb, mint a második " + szo2 + " szó.");

Console.WriteLine("Az első szó " + szo1.Length + " karakter hosszú, a második szó " + szo2.Length + " karakter hosszú.");

}

else if(szo2.Length < szo1.Length)

{

Console.WriteLine("A második szó " + szo2 + " rövidebb, mint az első " + szo1 + " szó.");

Console.WriteLine("A második szó " + szo2.Length + " karakter hosszú, az első szó " + szo1.Length + " karakter hosszú.");

}

else

{

Console.WriteLine("A két szó egyforma hosszú.");

Console.WriteLine("Az első szó is " + szo1.Length + " karakter hosszú, a második szó is " + szo2.Length + " karakter hosszú.");

}

Console.ReadKey();

}

}

}

# 049. azonosakeaszavak - Két szó közül eldönti, hogy azonosak-e

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_049.azonosakeaszavak

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Kérek egy szót! ");

string szo1 = Convert.ToString(Console.ReadLine());

Console.WriteLine("Kérek még egy szót és kiírom, hogy a szavak azonosak-e! ");

string szo2 = Convert.ToString(Console.ReadLine());

if (szo1 == szo2)

{

Console.WriteLine("A két szó egyforma.");

}

/\*else if (szo1 != szo2)

{

Console.WriteLine("A két szó nem egyforma.");

}

\*/

else

{

Console.WriteLine("A két szó nem egyforma.");

}

Console.ReadKey();

}

}

}

# 050. ahosszabbikszo - Két szó közül a hosszabbikat írja ki

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_050.ahosszabbikszo

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Kérek egy szót! ");

string szo1 = Convert.ToString(Console.ReadLine());

Console.WriteLine("Kérek még egy szót és kiírom a hosszabbikat! ");

string szo2 = Convert.ToString(Console.ReadLine());

if (szo1.Length > szo2.Length)

{

Console.WriteLine("Az első szó " + szo1 + " hosszabb, mint a második " + szo2 + " szó.");

Console.WriteLine("Az első szó " + szo1.Length + " karakter hosszú, a második szó " + szo2.Length + " karakter hosszú.");

}

else if (szo2.Length > szo1.Length)

{

Console.WriteLine("A második szó " + szo2 + " hosszabb, mint az első " + szo1 + " szó.");

Console.WriteLine("A második szó " + szo2.Length + " karakter hosszú, az első szó " + szo1.Length + " karakter hosszú.");

}

else

{

Console.WriteLine("A két szó egyforma hosszú.");

Console.WriteLine("Az első szó is " + szo1.Length + " karakter hosszú, a második szó is " + szo2.Length + " karakter hosszú.");

}

Console.ReadKey();

}

}

}

# 051. ketszoazonoskarakterei - Két szó azonos karakterei (hiányos!)

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

namespace \_051.ketszoazonoskarakterei

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Kérek egy szót! ");

string szo1 = Convert.ToString(Console.ReadLine());

Console.WriteLine("Kérek még egy szót és kiírom a kettő azonos karaktereit! ");

string szo2 = Convert.ToString(Console.ReadLine());

// Változók létrehozása

string betu\_lista = Console.ReadLine();

int db = 0;

// Tömb feltöltése

for (int i = 0; i < betu\_lista.Length; i++)

{

if(betu\_lista[i] == 'a')

{

db++;

}

}

Console.WriteLine("Azonos betűk a két szóban: " + betu\_lista);

Console.ReadKey();

}

}

}

# 059. TombosFeladat - Tömbös feladat - Sebesség

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace TombosFeladat

{

class Program

{

static void Main(string[] args)

{

//tömb deklarálása

int[] sebessegtomb = new int[10];

//feltöltés random értékekkel

Random r = new Random();

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

{

sebessegtomb[i] = r.Next(0, 200);

Console.Write(sebessegtomb[i] + " ");

}

Console.WriteLine();

//összegzés tétele: összegezzük a tömb elemeit!

int osszeg = 0;

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

{

osszeg += sebessegtomb[i];

}

Console.WriteLine("A tömb elemeinek összege: " + osszeg);

//Feltételes összegzés

int osszegf = 0;

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

{

if (sebessegtomb[i] > 100)

{

osszegf += sebessegtomb[i];

}

}

Console.WriteLine("A 100-nál gyorsabb kocsik sebességösszegei: " + osszegf);

//megszámlálás

int db = 0;

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

{

if (sebessegtomb[i] % 2 == 0)

{

db++;

}

}

Console.WriteLine("A páros számok: " + db);

//eldontés tétele

bool vane = false;

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

{

if (sebessegtomb[i] > 180)

{

vane = true;

//break;

}

Console.WriteLine(vane);

}

//minimum

int min = sebessegtomb[0];

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

{

if (sebessegtomb[i] < min)

{

min = sebessegtomb[i];

}

}

Console.WriteLine("A tömb elemeinek minimuma: " + min);

//maximum

int max = sebessegtomb[0];

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

{

if (sebessegtomb[i] > max)

{

max = sebessegtomb[i];

}

}

Console.WriteLine("A tömb elemeinek maximuma: " + max);

//csere1

int a = 4;

int b = 7;

Console.WriteLine("a: " + a + " b: " + b);

int temp = a;

a = b;

b = temp;

Console.WriteLine("a: " + a + " b: " + b);

Console.ReadLine();

}

}

}

# 065. Fajlkezeles - Fájlkezelés

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

namespace Fajlkezeles

{

class Program

{

static void Main(string[] args)

{

// Fájl beolvasás:

StreamReader sr = new StreamReader("elso.txt", Encoding.Default);

while (!sr.EndOfStream)

{

int szam =Convert.ToInt32( sr.ReadLine());

Console.WriteLine(szam);

}

sr.Close();

// Fájlírás:

StreamWriter sw = new StreamWriter("mentes.txt",true,encoding:Encoding.UTF8);

sw.WriteLine("Szasz Pisti!");

sw.WriteLine("Hogy vagy?");

sw.WriteLine("Érdekes fejlemények.");

sw.Close();

Console.ReadLine();

}

}

}

# 066. gamerek - Játékosok

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

namespace gamerek

{

class gamer

{

public string nev;

public string jelszo;

public string email;

public int pontszam;

}

class Program

{

static void Main(string[] args)

{

List<gamer> jatekosok = new List<gamer>();

StreamReader sr = new StreamReader("gamer.txt");

while(!sr.EndOfStream)

{

gamer gr = new gamer();

string sor = sr.ReadLine();

string[] darabok = sor.Split(';');

gr.nev = darabok[0];

gr.jelszo = darabok[1];

gr.email = darabok[2];

gr.pontszam = Convert.ToInt32(darabok[3]);

jatekosok.Add(gr);

}

sr.Close();

foreach(var item in jatekosok)

{

Console.WriteLine(item.email);

}

Console.ReadLine();

}

}

}

# 067. gamerek2 - Másik gamer osztály - Négy játékos és pontjaik

# Class1.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace gamerek2

{

class Class1

{

public string nev;

public string jelszo;

public string email;

public int pontszam;

}

}

# Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

namespace gamerek2

{

class gamer

{

public string nev;

public string jelszo;

public string email;

public int pontszam;

}

class Program

{

static void Main(string[] args)

{

List<gamer> jatekoslista = new List<gamer>();

// gamer létrehozása

gamer g = new gamer();

gamer g1 = new gamer();

gamer g2 = new gamer();

gamer g3 = new gamer();

g.nev = "Pisti";

g.jelszo = "qwert";

g.email = "hgfg@jhgf.hu";

g.pontszam = 57;

g1.nev = "Eszter";

g1.jelszo = "esztike";

g1.email = "esztike@gmail.hu";

g1.pontszam = 87;

g2.nev = "Gyula";

g2.jelszo = "gyuszko";

g2.email = "gyuszko@gmail.com";

g2.pontszam = 97;

g3.nev = "Fémember";

g3.jelszo = "metalman";

g3.email = "metalman@freemail.hu";

g3.pontszam = 77;

jatekoslista.Add(g);

jatekoslista.Add(g1);

jatekoslista.Add(g2);

jatekoslista.Add(g3);

/\*for (int i = 0; i < jatekoslista.Count; i++)

{

Console.WriteLine(jatekoslista[i].nev + " " + jatekoslista[i].pontszam);

}\*/

foreach (var item in jatekoslista)

{

Console.WriteLine(item.nev + " " + item.pontszam+ " pontja van.");

}

Console.ReadLine();

}

}

}

# 068. emberosztaly - 4 ember neve, életkora, a legidősebb és a legfiatalabb neve és életkora

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace emberosztaly

{

class ember

{

public string nev;

public int kor;

public string nem;

public double telefonszam;

public string lakcim;

}

class Program

{

static void Main(string[] args)

{

List<ember> emberlista = new List<ember>();

// ember létrehozása

ember e = new ember();

ember e1 = new ember();

ember e2 = new ember();

ember e3 = new ember();

e.nev = "Pisti";

e.kor = 26;

e.nem = "férfi";

e.telefonszam = 06305584057;

e.lakcim = "Budapest";

e1.nev = "Gyula";

e1.kor = 50;

e1.nem = "férfi";

e1.telefonszam = 06305569047;

e1.lakcim = "Miskolc";

e2.nev = "Eszter";

e2.kor = 28;

e2.nem = "nő";

e2.telefonszam = 06702865566;

e2.lakcim = "Debrecen";

e3.nev = "Julcsi";

e3.kor = 19;

e3.nem = "nő";

e3.telefonszam = 06202863948;

e3.lakcim = "Győr";

emberlista.Add(e);

emberlista.Add(e1);

emberlista.Add(e2);

emberlista.Add(e3);

/\*for (int i = 0; i < emberlista.Count; i++)

{

Console.WriteLine(emberlista[i].nev + " " + emberlista[i].kor);

}\*/

foreach (var item in emberlista)

{

Console.WriteLine(item.nev + " " + item.kor + " éves.");

}

ember maxkor = emberlista[0];

ember minkor = emberlista[0];

foreach (var item in emberlista)

{

if (item.kor > maxkor.kor)

{

maxkor = item;

}

}

foreach (var item in emberlista)

{

if (item.kor < minkor.kor)

{

minkor = item;

}

}

Console.WriteLine(maxkor.nev + " a legidősebb, " +maxkor.kor+ " éves.");

Console.WriteLine(minkor.nev + " a legfiatalabb, " + minkor.kor + " éves.");

Console.ReadLine();

}

}

}

# 069. helsinki1952 - Olimpia, érmek, helyezések stb.

# eredmeny.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace helsinki1952

{

class eredmeny

{

public int helyezes;

public int letszam;

public string kategoria;

public string sportag;

}

}

# Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

namespace helsinki1952

{

class Program

{

static List<eredmeny> eredmenylista = new List<eredmeny>();

static void Main(string[] args)

{

StreamReader sr = new StreamReader("helsinki.txt", encoding: Encoding.UTF7);

while(!sr.EndOfStream)

{

string sor = sr.ReadLine();

string[] darabok = sor.Split(' ');

eredmeny e = new eredmeny();

e.helyezes = Convert.ToInt32(darabok[0]);

e.letszam = Convert.ToInt32(darabok[1]);

e.kategoria = darabok[2];

e.sportag = darabok[3];

eredmenylista.Add(e);

}

sr.Close();

// adatok

/\*foreach(var item in eredmenylista)

{

Console.WriteLine(item.letszam);

}\*/

// 3. feladat

Console.WriteLine("3. feladat: ");

Console.WriteLine("A pontszerző helyezések száma: " +eredmenylista.Count);

// 4. feladat

Console.WriteLine("4. feladat:");

int aranydb = 0;

int ezustdb = 0;

int bronzdb = 0;

foreach(var item in eredmenylista)

{

if(item.helyezes==1)

{

aranydb++;

}

if (item.helyezes == 2)

{

ezustdb++;

}

if (item.helyezes == 3)

{

bronzdb++;

}

}

Console.WriteLine("aranydb: " +aranydb);

Console.WriteLine("ezustdb: " + ezustdb);

Console.WriteLine("bronzdb: " + bronzdb);

Console.WriteLine("összesen: " +(aranydb+ezustdb+bronzdb));

// 5. feladat

Console.WriteLine("5. feladat: ");

int pontok = 0;

foreach(var item in eredmenylista)

{

if (item.helyezes == 1)

{

pontok += 7;

}

if (item.helyezes == 2)

{

pontok += 5;

}

if (item.helyezes == 3)

{

pontok += 4;

}

if (item.helyezes == 4)

{

pontok += 3;

}

if (item.helyezes == 5)

{

pontok += 2;

}

if (item.helyezes == 6)

{

pontok += 1;

}

}

Console.WriteLine("A pontok száma: " +pontok);

// 6. feladat

Console.WriteLine("6. feladat");

int uszasdb = 0;

int tornadb = 0;

foreach(var item in eredmenylista)

{

if (item.kategoria== "uszas")

{

uszasdb++;

}

if (item.kategoria == "torna")

{

tornadb++;

}

}

// kiírás eldöntése

if(uszasdb>tornadb)

{

Console.WriteLine("Az úszás sportágban szereztek több érmet.");

}

if (uszasdb < tornadb)

{

Console.WriteLine("A torna sportágban szereztek több érmet.");

}

if (uszasdb == tornadb)

{

Console.WriteLine("Az úszás sportágban és a torna sportágban ugyanannyi érmet szereztek.");

}

// 7. feladat

Console.WriteLine("7. feladat");

StreamWriter sw = new StreamWriter("helsinki2.txt");

foreach(var item in eredmenylista)

{

int pont = 7 - item.helyezes;

if(pont==6)

{

pont = 7;

}

string segedkategoria = item.kategoria;

if(item.kategoria=="kajakkenu")

{

segedkategoria = "kajak-kenu";

}

sw.WriteLine(item.helyezes+" "+item.letszam+" "+pont+" "+segedkategoria+" "+item.sportag);

}

sw.Close();

// 8. feladat

Console.WriteLine("8. feladat: ");

eredmeny maxletszam = eredmenylista[0];

foreach (var item in eredmenylista)

{

if (item.letszam>maxletszam.letszam)

{

maxletszam = item;

}

}

Console.WriteLine("Helyezés: " +maxletszam.helyezes);

Console.WriteLine("Sportolók száma: " +maxletszam.letszam);

Console.WriteLine("Sportág: " +maxletszam.kategoria);

Console.WriteLine("Versenyszám: " +maxletszam.sportag);

Console.ReadLine();

}

}

}

# 071 domino - Dominó

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Domino

{

class domino

{

public int bal;

public int jobb;

public domino(int bal, int jobb)

{

this.bal = bal;

this.jobb = jobb;

}

}

class Program

{

static void Main(string[] args)

{

List<domino> dominolista = new List<domino>();

StreamReader sr = new StreamReader("domino.txt", Encoding.Default);

while (!sr.EndOfStream)

{

string[] darabok = sr.ReadLine().Split(' ');

dominolista.Add(new domino(Convert.ToInt32(darabok[0]), Convert.ToInt32(darabok[1])));

}

sr.Close();

foreach (var item in dominolista)

{

Console.WriteLine(" " + item.bal + " | " + item.jobb);

}

Console.Write(" 3.feladat: ");

Console.WriteLine(" Dominók száma:" + dominolista.Count + "db");

Console.Write(" Kérek egy számot 1-13-ig! ");

int sorszam = Convert.ToInt32(Console.ReadLine());

Console.Write(" 4.feladat: ");

Console.WriteLine(" A(z) " + sorszam + ". sorszámnak megfelelő dominó: " + dominolista[sorszam - 1].bal + " " + dominolista[sorszam - 1].jobb);

Console.Write(" 5.feladat: ");

int duplakszama = 0;

foreach (var item in dominolista)

{

if (item.bal == item.jobb)

{

duplakszama++;

}

}

Console.WriteLine(" Dupla dominók száma: " + duplakszama + "db");

Console.Write(" 6.feladat: ");

domino elozo\_domino = dominolista[0];

int szamlalo = 0;

bool szabalyose = true;

foreach (var item in dominolista)

{

if (szamlalo > 0)

{

if (item.bal != elozo\_domino.jobb)

{

szabalyose = false;

}

}

szamlalo++;

elozo\_domino = item;

}

if (szabalyose == true)

{

Console.WriteLine(" Szabályosak az illesztések");

}

else

{

Console.WriteLine(" Nem szabályosak az illesztések");

}

Console.WriteLine();

Console.WriteLine(" Vége a proginak sajnos. Indítsd újra, ha akarod! :) ");

Console.ReadKey();

}

}

}

# 072. sudokuCLI - Sudoku konzolos

# Osztaly.cs

using System;

public class Class1

{

class Feladvany

{

public string Kezdo { get; private set; }

public int Meret { get; private set; }

public Feladvany(string sor)

{

Kezdo = sor;

Meret = Convert.ToInt32(Math.Sqrt(sor.Length));

}

public void Kirajzol()

{

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

{

if (Kezdo[i] == '0')

{

Console.Write(".");

}

else

{

Console.Write(Kezdo[i]);

}

if (i % Meret == Meret - 1)

{

Console.WriteLine();

}

}

}

}

}

# Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO; // Ha fájl beolvasásról, vagy írásról van szó, ezt a névteret mindenképpen ide be kell írnunk.

namespace SudokuCLI

{

class Feladvany

{

public string Kezdo { get; private set; }

public int Meret { get; private set; }

public Feladvany(string sor) // Ez egy konstruktor, ami átvesz egy sor nevű változót, amit a feladvanyok.txt fájlunkból kapunk. Ezt a sor nevű változót, mint látni, odaadja a francia zárójelben lévő Kezdo nevű mezőnek. A feladvanyok.txt fájlból érkező sor egy kezdő állapot, azért is lett Kezdo a neve, amit el kell tudnunk tárolni egy változóban, ami a sor nevű változó lesz, ott fog tárolódni az az adatsor, ami a fájlból jön. Azonkívül itt ezzel a konstruktorral egy példányt készítünk a feladvanyok.txt fájlból, példányosítunk, hogy a felhasználó számára is hozzáférhetővé váljanak a benne szereplő adatok, hogy lássa magyarul a sudoku feladványait kirajzolva a képernyőre.

{

Kezdo = sor;

Meret = Convert.ToInt32(Math.Sqrt(sor.Length)); // Itt az történik, hogy a fájlból beolvasott sor hosszából (sor.Length) négyzetgyököt vonunk (Math.Sqrt), átkonvertáljuk egész számmá (Convert.ToInt32) és ez lesz a sudoku mérete. Tehát, ha a fájlban az adott sorunk 16 karakterből áll, akkor abból vonunk négyzetgyököt, ami ugye 4, ez lesz a sudoku mérete, tehát egy 4x4-es négyzetben fog kirajzolódni. Ha 64 karakterből áll a sor, annak a négyzetgyöke 8, tehát 8x8-as lesz a sudoku mérete, ekkora négyzetet fog kirajzolni.

}

public void Kirajzol() // Ez a Kirajzol rész végzi gyakorlatilag a sudoku négyzeteinek kirajzolását. Mit is csinál? Végigmegy egy for ciklussal a fájlban található adott soron (Kezdo.Length), ha 0-át talál a sorban, akkor annak a helyére pontot rajzol (if(Kezdo[i] == 0) {Console.Write(".");}), ha nem nullát talál, akkor meg kiírja az ott szereplő számot (Console.Write(Kezdo[i]);). Kicsit másképp fogalmazva: Ha a Kezdo állapot i-edik, azaz akármelyik eleme 0, akkor pontot rajzol, különben ha a Kezdo i-edik, azaz akármelyik eleme nem nulla, akkor kiírja ezt az i-edik elemet, ami egy nullától különböző szám lesz, ami a fájlban szerepel.

{

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

{

if (Kezdo[i] == '0')

{

Console.Write(".");

}

else

{

Console.Write(Kezdo[i]);

}

if (i % Meret == Meret - 1)

{

Console.WriteLine();

}

}

}

}

class Program

{

static List<Feladvany> lista = new List<Feladvany>(); // Itt egy listát készítünk a Feladvany osztályban létrejött adatoknak a tárolására.

static void Main(string[] args)

{

// beolvassuk a progiba a feladvanyok.txt fájlt UTF8-as kódolással, hogy az ékezetes betűket is kezelni tudja és, hogy képes legyen a program dolgozni az adatokkal. Ha ez nem történik meg, akkor a progi nem tud mivel dolgozni. Ezért ezt a fájlt be kell illeszteni a program mappájába, a bin --> Debug mappába.

StreamReader sr = new StreamReader("feladvanyok.txt", Encoding.UTF8);

string sor = "";

while(!sr.EndOfStream)

{

sor = sr.ReadLine();

Feladvany f = new Feladvany(sor); // A fenti Feladvany nevű osztályból egy példány. Jobban mondva a feladvanyok.txt file egy sorából készített példány, ami a Feladvany nevű osztályban szerepel. Példányosítás. Ezt az új példányt f-nek neveztük el, ami már nem ugyanaz, mint a fenti Feladvany sor az f betű miatt. Mivel a fenti Feladvany osztály konstruktora egy sor nevű változót vár, az új példány zárójelébe a sor-t kell írni.

lista.Add(f); // Az f nevű Feladvany osztály példányainak elemeit a listámhoz adom.

}

sr.Close(); // Bezárom a beolvasott fájlomat, hogy az abban szereplő adatokat ne tudja átírni, piszkálni senki.

Console.WriteLine("3. feladat");

Console.WriteLine("Feladványok száma: " + lista.Count); // Ez a sor összeszámolja a beolvasott fájlban szereplő sorokat, amik a feladványok darabszáma lesz és azt írja ki. Jelen esetben 98.

Console.WriteLine("4. feladat");

int meret = 0;

do

{

Console.Write("Kérem a sudoku méretét!: ");

meret = Convert.ToInt32(Console.ReadLine());

}

while (meret<4 || meret>9);

int meretDB = 0;

for (int i = 0; i < lista.Count; i++)

{

if(lista[i].Meret==meret)

{

meretDB++;

}

}

Console.WriteLine("Ennyi van a megadott méretből: " +meretDB);

Console.WriteLine("{0}x{0} méretű feladványból {1} darab van tárolva. ", meret, meretDB);

Console.WriteLine("5. feladat");

Random r = new Random();

int kivalasztottIndex = 0;

do

{

kivalasztottIndex = r.Next(0, lista.Count);

}

while (lista[kivalasztottIndex].Meret != meret);

Console.WriteLine("A kiválasztott feladvány kezdő állapota: " + lista[kivalasztottIndex].Kezdo);

Console.WriteLine("6. feladat");

int nemNullaDB = 0;

int hossz = lista[kivalasztottIndex].Kezdo.Length;

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

{

if(lista[kivalasztottIndex].Kezdo[i]!='0')

{

nemNullaDB++;

}

}

double kitoltottseg = (double)nemNullaDB / hossz \* 100;

Console.WriteLine("A kiválasztott feladvány kitöltöttsége: " + kitoltottseg + "%");

Console.WriteLine("7. feladat");

Console.WriteLine("A kiválasztott feladvány kirajzolva: ");

lista[kivalasztottIndex].Kirajzol();

Console.WriteLine("8. feladat");

string fajlneve = "sudoku" + meret + ".txt";

StreamWriter sw = new StreamWriter(fajlneve);

for (int i = 0; i < lista.Count; i++)

{

if(lista[i].Meret==meret)

{

sw.WriteLine(lista[i].Kezdo);

}

}

sw.Close();

Console.WriteLine("A kiválasztott méretűek kiírva a fájlba.");

Console.WriteLine("A " + fajlneve + " állomány létrehozva ennyi sorral: " +meretDB);

Console.ReadKey();

}

}

}

# 074. IskolaGUI - Iskola tanulóinak listája és adataik

# Form1.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

namespace IskolaGUI

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

StreamReader sr = new StreamReader("nevekGUI.txt", Encoding.UTF8);

string sor = "";

while(!sr.EndOfStream)

{

sor = sr.ReadLine();

listBox1.Items.Add(sor);

}

sr.Close();

}

private void button1\_Click(object sender, EventArgs e)

{

if(listBox1.SelectedIndex==-1) // A -1 egy azt jelzi, hogy nincs a listBox1-ben kijelölve semmi.

{

MessageBox.Show("Nem jelölt ki tanulót!");

}

else

{

int hanyadik = listBox1.SelectedIndex;

listBox1.Items.RemoveAt(hanyadik);

}

}

private void button2\_Click(object sender, EventArgs e)

{

try

{

StreamWriter sw = new StreamWriter("nevekNEW.txt"); //("c:\\valami\\nevekNEW.txt");

foreach(var elem in listBox1.Items)

{

sw.WriteLine(elem);

}

sw.Close();

MessageBox.Show("Sikeres mentés.");

}

catch(Exception hiba)

{

MessageBox.Show(hiba.Message);

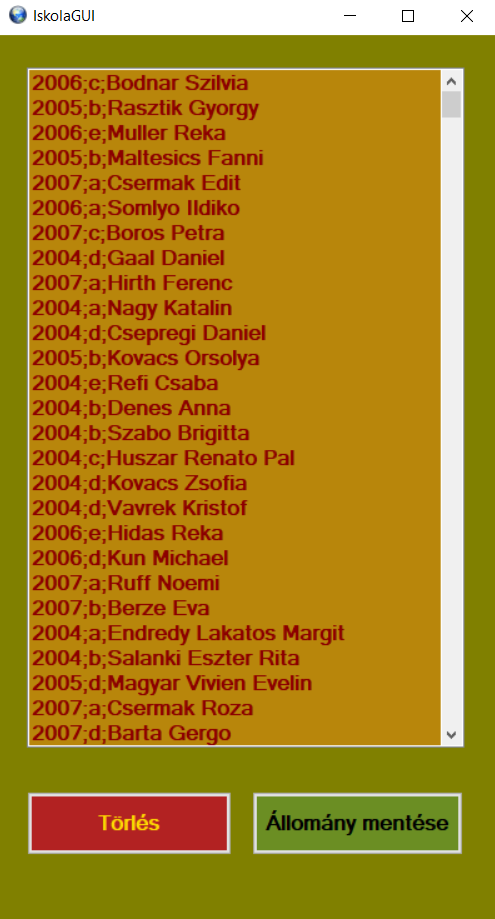
}

}

}

}

**A program futás közben:**



# Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace IskolaGUI

{

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

# 075. fizetes - Elégedett vagy-e a fizetéseddel?

# Form1.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace fizetes

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

label1.Text = "Hát elég nagy hülye vagy!";

}

private void button2\_Click(object sender, EventArgs e)

{

label1.Text = "Akkor megemelem 100 ezerrel.";

}

private void button2\_MouseHover(object sender, EventArgs e)

{

// label1.Text = "Minek piszkálod?"

Random r = new Random();

int x = r.Next(1,this.Width-button2.Width);

int y = r.Next(1, this.Height-button2.Height);

button2.SetBounds(x,y,button2.Width, button2.Height);

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void label1\_Click(object sender, EventArgs e)

{

}

}

}

# Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace fizetes

{

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

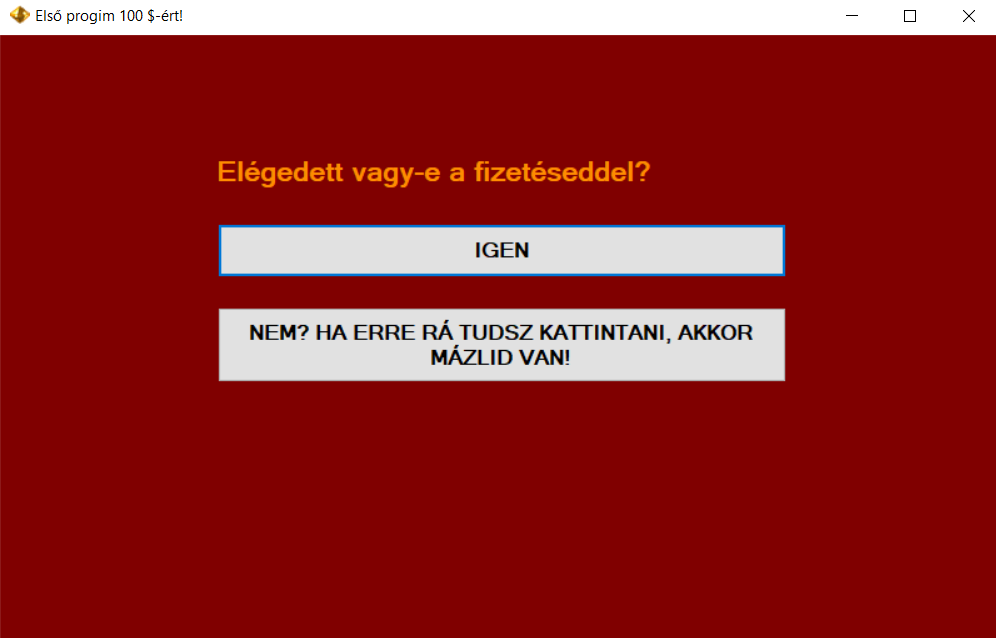
Application.Run(new Form1());

}

}

}

# A program futás közben:



# 076. Paint alapok

# Form1.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace paint

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

trackBar1.Minimum = 1;

trackBar1.Maximum = 100;

trackBar2.Minimum = 1;

trackBar2.Maximum = 100;

}

private void button1\_Click(object sender, EventArgs e)

{

Graphics gh = pictureBox1.CreateGraphics();

Pen p = new Pen(Color.Red);

Pen p2 = new Pen(Color.Green);

Pen p3 = new Pen(Color.Blue);

Pen p4 = new Pen(Color.DarkMagenta);

gh.DrawLine(p,10,10,100,100);

Random r = new Random();

int korx = r.Next(10, 150);

int kory = r.Next(10, 150);

gh.DrawEllipse(p2,korx,kory,40,30);

gh.DrawRectangle(p3, kory, korx,100,40);

gh.DrawLine(p4,kory, korx, 160, 50);

}

private void button2\_Click(object sender, EventArgs e)

{

colorDialog1.ShowDialog();

}

private void button3\_Click(object sender, EventArgs e)

{

Graphics gh = pictureBox1.CreateGraphics();

gh.Clear(Color.Wheat);

}

private void button4\_Click(object sender, EventArgs e)

{

Graphics gh = pictureBox1.CreateGraphics();

Brush br = new SolidBrush(Color.Red);

Brush br2 = new SolidBrush(Color.DarkMagenta);

Brush br3 = new SolidBrush(Color.DarkOrange);

Brush br4 = new SolidBrush(Color.DarkCyan);

gh.FillRectangle(br,20,20,120,120);

gh.FillEllipse(br2, 30, 30, 130, 130);

gh.FillRectangle(br3, 60, 60, 150, 110);

gh.FillEllipse(br4, 80, 80, 190, 130);

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

private void pictureBox1\_MouseDown(object sender, MouseEventArgs e)

{

egergomb = true;

}

private void pictureBox1\_MouseHover(object sender, EventArgs e)

{

}

private void pictureBox1\_MouseMove(object sender, MouseEventArgs e)

{

if(egergomb)

{

Graphics gh = pictureBox1.CreateGraphics();

Pen p2 = new Pen (colorDialog1.Color);

int x = e.X;

int y = e.Y;

gh.DrawEllipse(p2, x, y, trackBar1.Value, trackBar2.Value);

}

}

bool egergomb = false;

private void pictureBox1\_MouseUp(object sender, MouseEventArgs e)

{

egergomb = false;

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void trackBar1\_Scroll(object sender, EventArgs e)

{

}

private void trackBar2\_Scroll(object sender, EventArgs e)

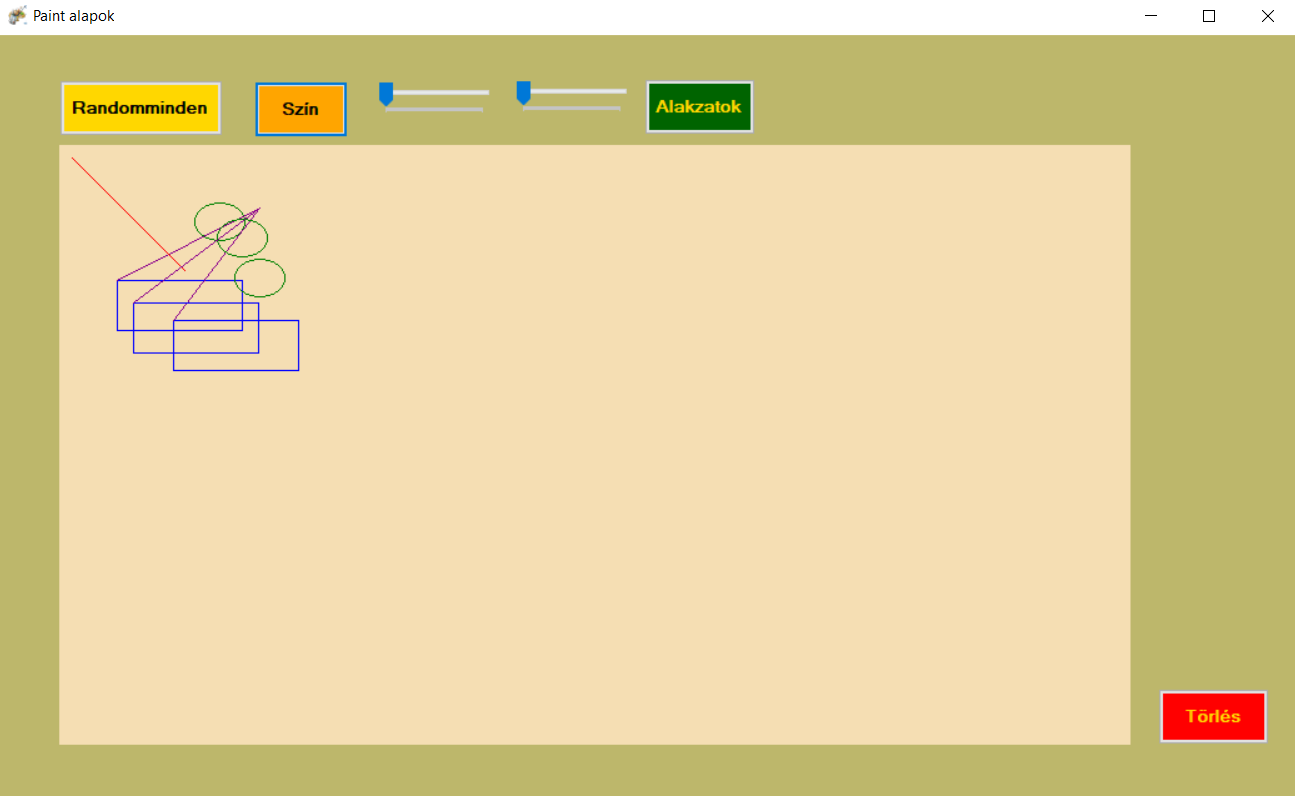
{

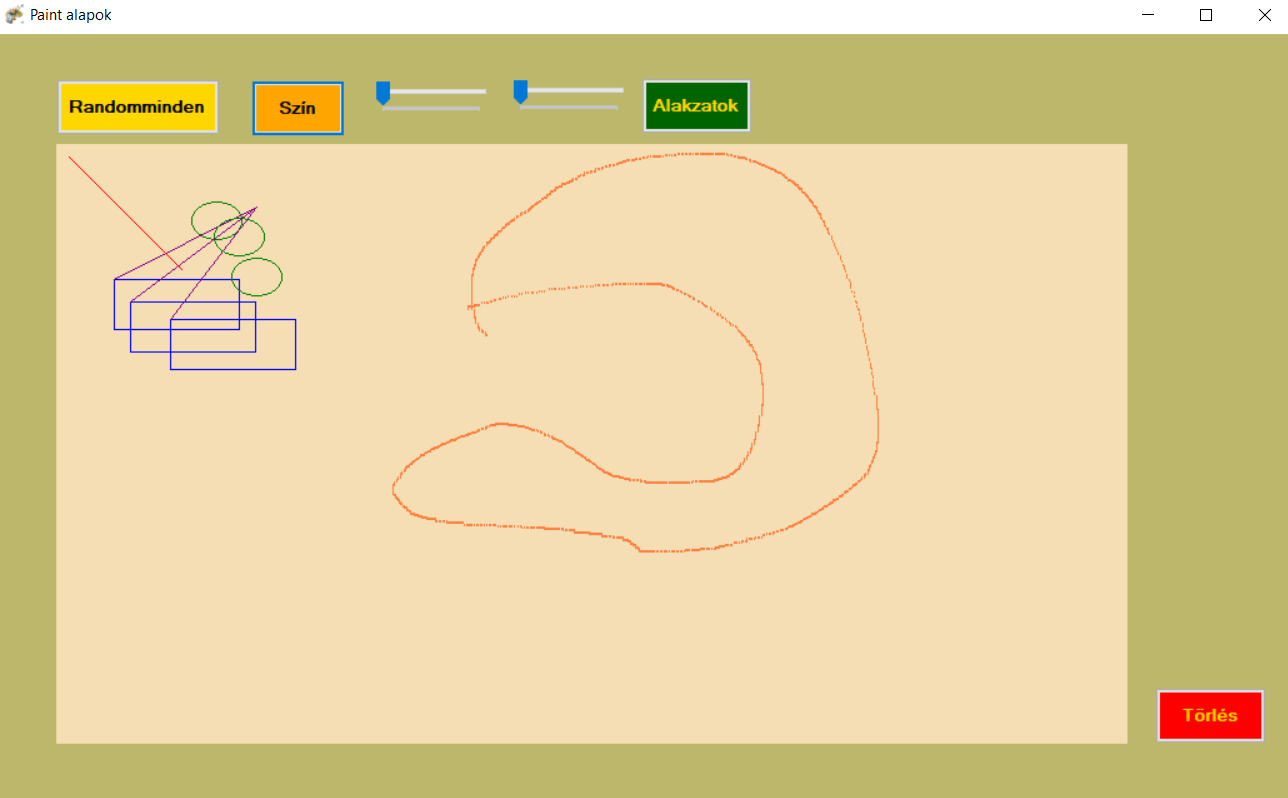
}

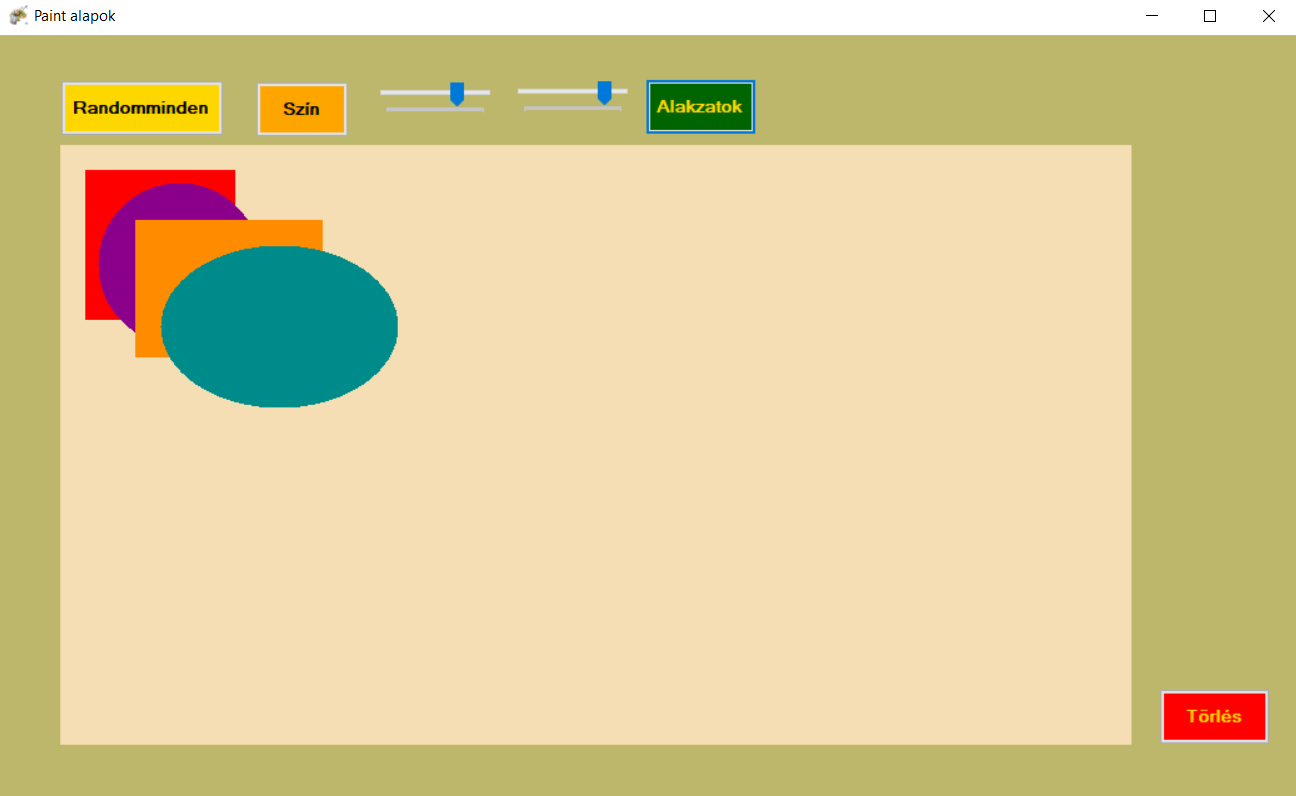
}

}

# A program futás közben:







# 084. godrok - Gödrök adatai (CLI-s, konzolos feladat)

# Program.cs

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Godrok

{

internal class Program

{

static int[] melysegek = new int[2000];

static int darab = 0;

static int felhasznalo;

static void Main(string[] args)

{

Feladat1();

Feladat2();

Feladat3();

Feladat4();

Feladat5();

Feladat6();

}

static void Feladat1()

{

Console.WriteLine("1. feladat: ");

StreamReader reader = new StreamReader("melyseg.txt");

string line = reader.ReadLine();

while (line != null)

{

melysegek[darab] = Int32.Parse(line);

darab++;

line = reader.ReadLine();

}

reader.Close();

Console.WriteLine($"A fájl adatainak száma: {darab}");

Console.WriteLine();

//Console.ReadLine();

}

static void Feladat2()

{

Console.WriteLine("2. feladat: ");

Console.Write("Adjon meg egy távolságértéket! ");

felhasznalo = Int32.Parse( Console.ReadLine() );

felhasznalo--;

Console.WriteLine($"Ezen a helyen a felszín { melysegek[felhasznalo] } méter mélyen van.");

Console.WriteLine();

//Console.ReadLine();

}

static void Feladat3()

{

Console.WriteLine("3. feladat: ");

int nullak = 0;

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

{

if (melysegek[i] == 0)

{

nullak++;

}

}

double erintetlen = (double)nullak / (double)darab \*100;

erintetlen = Math.Round(erintetlen, 2);

//Console.WriteLine(erintetlen);

Console.WriteLine($"Az érintetlen terület aránya {erintetlen} %.");

//Console.WriteLine(nullak);

Console.WriteLine();

//Console.ReadLine();

}

static void Feladat4()

{

StreamWriter writer = new StreamWriter("godrok.txt");

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

{

if(melysegek[i] != 0)

{

writer.Write( melysegek[i] +" ");

}

else

{

if(melysegek[i -1] != 0)

{

writer.WriteLine();

}

}

}

writer.Close();

//Console.ReadLine();

}

static void Feladat5()

{

Console.WriteLine("5. feladat: ");

int godrok = 0;

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

{

if(melysegek[i] != 0 && melysegek[i -1]==0)

{

godrok++;

}

}

Console.WriteLine($"A gödrök száma: {godrok}");

Console.WriteLine();

//Console.ReadLine();

}

static void Feladat6()

{

Console.WriteLine("6. feladat: ");

if(melysegek[felhasznalo] != 0)

{

Feladat6a();

Feladat6b();

Feladat6c();

Feladat6d();

Feladat6e();

}

else

{

Console.WriteLine("Az adott helyen nincs gödör.");

}

Console.WriteLine();

}

static int kezdete;

static int vege;

static int terfogat;

static void Feladat6a()

{

Console.WriteLine("a)");

kezdete = felhasznalo;

vege = felhasznalo;

while(melysegek [kezdete -1] != 0)

{

kezdete--;

}

while(melysegek[vege +1] != 0)

{

vege++;

}

Console.WriteLine($"A gödör kezdete {kezdete +1} méter, a gödör vége {vege +1} méter.");

//Console.ReadLine();

}

static void Feladat6b()

{

Console.WriteLine("b)");

int x = kezdete;

int y = vege;

while (melysegek [x] <= melysegek[x +1])

{

x++;

}

while (y > x && melysegek [y] <= melysegek[y -1])

{

y--;

}

if (x == y)

{

Console.WriteLine("Folyamatosan mélyül.");

}

else

{

Console.WriteLine("Nem mélyül folyamatosan.");

}

Console.WriteLine();

//Console.ReadLine();

}

static void Feladat6c()

{

Console.WriteLine("c)");

int max = 0;

for (int i = kezdete; i <= vege; i++)

{

if (melysegek[i] > max)

{

max = melysegek[i];

}

}

Console.WriteLine($"A legnagyobb mélysége {max} méter.");

//Console.ReadLine();

}

static void Feladat6d()

{

Console.WriteLine("d)");

terfogat = 0;

for (int i = kezdete; i <= vege; i++)

{

terfogat += melysegek[i];

}

terfogat \*= 10;

Console.WriteLine($"A térfogata {terfogat} m^3.");

//Console.ReadLine();

}

static void Feladat6e()

{

Console.WriteLine("e)");

int viz = terfogat - ((vege - kezdete + 1) \* 10);

Console.WriteLine($"A vízmennyiség {viz} m^3.");

Console.ReadLine();

}

}

}

# 092. Tesztverseny - Konzolos

# Program.cs

using System;

using System.IO;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Security.AccessControl;

namespace tesztverseny

{

internal class Program

{

struct versenyrekord

{

public string kod;

public string valasz;

public int pont;

}

static versenyrekord[] verseny = new versenyrekord[500];

static int db = 0;

static string jovalasz = "";

static string azonosito = "";

static int hanyadik = 0;

static void Main(string[] args)

{

feladat1();

feladat2();

feladat3();

feladat4();

feladat5();

feladat6();

feladat7();

Console.WriteLine("A program gombnyomásra vár!");

Console.ReadKey();

}

static void feladat1()

{

Console.WriteLine("1. feladat: Az adatok beolvasása");

Console.WriteLine();

FileStream fajlbe = new FileStream("..\\..\\valaszok.txt", FileMode.Open);

StreamReader beolvas = new StreamReader(fajlbe);

jovalasz = beolvas.ReadLine();

while (!beolvas.EndOfStream)

{

string[] tordel = beolvas.ReadLine().Split(' ');

verseny[db].kod = tordel[0];

verseny[db].valasz = tordel[1];

db++;

}

beolvas.Close();

fajlbe.Close();

}

static void feladat2()

{

Console.WriteLine("2. feladat: A vetélkedőn {0} versenyző indult.", db);

Console.WriteLine();

}

static void feladat3()

{

Console.Write("3. feladat: A versenyző azonosítója= ");

azonosito = Console.ReadLine();

int i = 0;

while (azonosito != verseny[i].kod && i<db)

{

i++;

}

if (i<db)

{

Console.WriteLine("{0} (a versenyző válasza)", verseny[i].valasz);

i = hanyadik;

}

else

{

Console.WriteLine("Nincs ilyen kóddal versenyző.");

}

Console.WriteLine();

}

static void feladat4()

{

Console.WriteLine("4. feladat: ");

Console.WriteLine("{0} (a helyes megoldás)", jovalasz);

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

{

if (jovalasz[i] == verseny[hanyadik].valasz[i])

{

Console.Write("+");

}

else

{

Console.Write(' ');

}

}

Console.WriteLine(" (a versenyző helyes válaszai)");

Console.WriteLine();

}

static void feladat5()

{

Console.Write("5. feladat: A feladat sorszáma = ");

int sorszam = Convert.ToInt32(Console.ReadLine())-1;

int helyes = 0;

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

{

if (jovalasz[sorszam] == verseny[i].valasz[sorszam])

{

helyes++;

}

}

double szazalek = Convert.ToDouble(helyes)/Convert.ToDouble(db) \* 100;

Console.WriteLine("A feladatra {0} fő, a versenyzők {1}%-a adott helyes választ.", helyes, szazalek);

Console.WriteLine();

}

static void feladat6()

{

Console.WriteLine("6. feladat: A versenyzők pontszámának meghatározása");

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

{

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

{

if (jovalasz[j] == verseny[i].valasz[j])

{

if (j >= 0 && j <= 4)

{

verseny[i].pont += 3;

}

if (j >= 5 && j <= 9)

{

verseny[i].pont += 4;

}

if (j >= 10 && j <= 12)

{

verseny[i].pont += 5;

}

if (j == 13)

{

verseny[i].pont += 6;

}

}

}

}

FileStream fajlki = new FileStream("..\\..\\pontok.txt", FileMode.Create);

StreamWriter kiir = new StreamWriter(fajlki);

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

{

kiir.WriteLine("{0} {1}", verseny[i].kod, verseny[i].pont);

}

kiir.Close();

fajlki.Close();

Console.WriteLine();

}

static void feladat7()

{

Console.WriteLine("7. feladat: A verseny legjobbjai:");

Console.WriteLine();

// Rendezzük pontszám szerint csökkenő sorrendben a versenyzők pontszámait. Legjobbak legelöl.

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

{

for (int j = 0; j < db-i; j++)

{

if (verseny[j].pont <= verseny[j+1].pont)

{

versenyrekord csere = verseny[j];

verseny[j] = verseny[j+1];

verseny[j+1] = csere;

}

}

}

// Ellenőrző kiíratás

/\*for (int i = 0;i < db; i++)

{

Console.WriteLine("{0} {1}", verseny[i].kod, verseny[i].pont);

Console.ReadKey();

}\*/

int helyezes = 1;

Console.WriteLine("{0}. díj ({1} pont): {2}", helyezes, verseny[0].pont, verseny[0].kod);

int k = 0;

do

{

k++;

if (verseny[k].pont == verseny[k-1].pont)

{

Console.WriteLine("{0}. díj ({1} pont): {2}", helyezes, verseny[k].pont, verseny[k].kod);

}

if (verseny[k].pont < verseny[k - 1].pont)

{

helyezes++;

Console.WriteLine("{0}. díj ({1} pont): {2}", helyezes, verseny[k].pont, verseny[k].kod);

}

}

while (helyezes < 3 || k == db);

}

}

}