*c# DOLGOZAT*

***TÖMBÖK***

***2023***

|  |  |  |
| --- | --- | --- |
| ***Osztályzat*** | ***Feltétel*** | ***Pontok*** |
| ***1*** | 0 – 39% | ***0 - 19*** |
| ***2*** | 40 – 55% | ***20 - 27*** |
| ***3*** | 56 – 75% | ***28 - 37*** |
| ***4*** | 75 - 89% | ***38 - 44*** |
| ***5*** | 90 -100% | ***45 - 51*** |

**A dolgozat kidolgozásához készítsen egy mappát az asztalon a saját neve (Vezetéknév Keresztnév). Ebbe a mappába másolja át a \_dolgozat.docx-t a \\optimus\h\Vastag Atila\Leadando hálózati meghajtóról. Ebben találja a kidolgozandó feladato-kat.**

**A feladat gyakorlati részéhez az asztalon elkészített mappába készítsen egy feladat nevű mappát. Ebben a mappában oldja meg a dolgozat gyakorlati részét.**

**A dolgozatot leadásakor az asztalon elkészített mappát másolja fel az \\optimus\h\Vastag Atila\Leadando hálózati meghajtóra.**

**1.** Mi a tömb? (**1**)

**válasz: Homogén adatszerkezet ahol az elemeknek a helyét indexek azonosítják**

**2.** Egy tömb csak homogén elemekből állhat. Mit jelent ez? (**1**)

**válasz: Minden elemnek azonos típusúnak kell lennie**

**3.** Hogyan tudjuk azonosítani tömbön belül az egyes elemek értékét? (**2**)

**válasz: Indexekkel, *tömbnév[index]* ez mindig az index+1-edik számot adja vissza**

|  |  |  |  |
| --- | --- | --- | --- |
| A *tömb neve* | 0 *index* | 1 *index* | 2 *index* |
| 3 *érték* | -2 *érték* | 0 *érték* |

**4.** Jelöld be a tömb nevét, indexe(ke)t és az értéke(ke)t! (**3**)

**5.** A negyedik feladatban látható tömböt használva írd a vonalra a megfelelő értékeket: (**3**)

A[1] = -2 A[2] = 0 A[0] = 3

**6.** Deklarálj egy: **(4)**

* 7 elemből álló char tipusú tömböt:

**char[] karakterTomb = new char[7]**

* 4 elemből álló string tipusú tömböt:

**string[] szovegTomb = new string[4]**

* 34 elemből álló double tipusú tömböt:

**double[] tizedesSzam = new double[34]**

* 10 elemből álló int tipusú tömböt:

**int[] szamTomb = new int[10]**

|  |  |  |  |
| --- | --- | --- | --- |
| A | 0 | 1 | 2 |
| 4 | ~~tej~~ 5 | 3~~,25~~ |

**7.** Adva van egy tömb. Ha van benne hiba javítsd ki! (**2**)

**8.** Adva van egy üres tömb, és a tömb elemei. Az elemeket írd a megfelelő helyre (pozicióra) a tömben. (**3**)  
  
A[5] = 63 A[0] =2 A[2] = -4 A[4] = -45 A[1] = 3 A[3] = -5

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | 0 | 1 | 2 | 3 | 4 | 5 |
| 2 | 3 | -4 | -5 | -45 | 63 |

**9.** Mit írna ki az alábbi programrészlet a képernyőre? **(3)**

int[] A = new int[] { 3, 4, 53, 23, 0, 53, 33, 87 };  
for (int i=1; i < A.Length; i+=2)

{

Console.WriteLine($"A[{i}] = {A[i]}");

}

**válasz: mivel 1-ről kezdődik és 2-vel ugrál ezért a következőt:**

**A[1] = 4**

**A[3] = 23**

**A[5] = 53**

**A[7] = 87**

**10.** Keresd meg a hibákat! (**5**)

int osszeg = ~~1~~; 0

int[] A = new int[10]

{

3, 4, 53, ~~’c’~~, 102, ~~”~~42~~”~~, 0, ~~”kettes”~~, 33~~.6~~, 87

};

for (int i=~~1~~; i > A.~~Count~~; i~~--~~) *for (int i = 0; i< A.Length; i++)*

{

~~osszeg[i] += A~~; *osszeg += A[i]*

}

ConsoleWriteLine($”A tomb elemeinek az osszege: ~~{A}~~”); *{osszeg}*

**BONUS**

Egy tömbön végig lehet lépkedni for és foreach ciklussal is. Mi a kettő közt a különbség és mikor melyik használható, előnyök és hátrányok? (**5**)  
**A for ciklus index még a foreach index nélkül megy végig a tömbön**

**Ha azt kell megtudnunk hogy hanyadik egy elem akkor előnyösebb a for**

**Kiíráshoz a foreach előnyösebb**

**GYAKORLATI FELADAT** (**28 + 2**)

Az iskolában üres aluminium doboz gyüjtésének versenyét szervezik. A szervezők Téged kértek meg, hogy készíts egy alkalmazást, amely a versenyhez szükséges statisztikákat számolja ki.

A szervezők a következő adatokkal dolgoznak:

* Osztály neve
* 0,25 dobozok száma
* 0,33 dobozok száma
* 0,5 dobozok száma

A dobozok különböző pontászámokat érnek:

* 0,25 → 1 pont
* 0,33 → 2 pont
* 0,5 → 3 pont

A programnak az elején be kell kérnie az induló osztályok számát. **(2)**

Az adatok feltöltése után **(5)** írjuk ki a képernyőre az adatokat pontszámok alapján csökkenő sorrendben (a rendezéshez használhat beépített függvényeket sem LINQ-t) semmi **(3)** az alábbi formátumban **(3)**:

*Osztály 0,25 dobozok száma 0,33 dobozok száma 0,5 dobozok száma Pontok*

*11b 15 14 55 153*

*10b 8 20 35 118*

Számítsuk ki az összesen összegyűjtött dobozok számát és a táblázat alá írjuk ki: **(2)**

*A részt vett osztályok összesen {dobozok száma} gyűjtöttek össze.*

Számítsuk ki az átlag pontszámot és írjuk ki: **(2)**

*A részt vett osztályok pontjainak az átlaga: {átlag}.*

Keressük ki, hogy hány osztály teljesített az átlag felett, és írjuk ki a minta alapján **(2):**

*A pontok alapján {átlag felett teljesítő osztályok száma} osztály teljesítette az átlag felett..*

Keressük ki, hogy mely doboz típusból gyűlt össze a legtöbb, és írjuk ki a minta alapján **(4):**

*A legtöbb a {doboz típusa} dobozból gyűlt össze.*

Keressük ki, hogy a lejobb és a legrosszabb osztály közt mekkora a pontkülönbség, és írjuk ki a minta alapján **(3):**

*A legtöbb és a legkevesebb pontot gyűjtő osztály közt {különbség} pontkülönbség van.*

Volt-e olyan osztály, amely pontosan 100 pontot ért el. A választ a mintának megfelelően írjuk ki **(2):**

*Van / Nincs olyan osztály, amely pontosan 100 pontot ért el.*

Ügyeljünk az íratlan szabályok (konvenciók) betartására: változó nevek, függvény nevek, osztály nevek, könyvtárak nevének írására. Az egyes egészeket kötelező függvényekbe szervezni! **(2)**