**Pontozási útmutató (minden feladathoz):**

**Hivatalból:**

* **1 pont (hivatalból)**
* **1 pont (ha nincs kompilálási hiba)**
* **1 pont (változók deklarálása)**
* **2 pont (helyes adatbevitel, a bevitt adatok helyességének ellenőrzése a feladatban megfogalmazott feltételeknek megfelelően)**
* **4 pont (helyes algoritmus megvalósítása: prímek, számjegyek, rendezés, karakter feldolgozás…)**
* **1 pont (az eredmény kiírása)**

**Egydimenziós tömbök:**

1. Olvass be a billentyűzetről két, **a** és **b** (a<b), egyenként legtöbb 9 jegyű természetes számot. Írasd a ***fibo.txt*** állományba a Fibonacci sorozat azon elemeit, amelyek az **[a, b]** intervallumban vannak.

**Példa:** ha a beolvasott számok **a=4, b=30**

A ***fibo.txt*** állomány tartalma: **5 8 13 21**

1. A ***numere.txt*** állomány egyetlen sora legtöbb 100 darab, egyenként legtöbb 4 jegyű természetes számot tartalmaz, szóközzel elválasztva. Írj programot, amely kiolvassa a számokat a ***numere.txt*** állományból és kiírja a képernyőre, szóközzel elválasztva, az állományban található összes nullától különböző természetes számot, növekvő sorrendben. Ha nem létezik ilyen szám, akkor a ***NU EXISTA*** üzenetet kell a képernyőre kiírni.

**Példa:** Ha a ***numere.txt*** állomány tartalma: **567 312  9 0 400**

A képernyőre kiírt számok: **9 312 400 567**

1. Olvass be a billentyűzetről egy **n** természetes számot (n>=100). Döntsd el az adott számról, hogy „*hegy-völgy*” szám-e. Egy természetes szám „*hegy-völgy*” szám, ha a számjegyei egy adott pozícióig növekvő sorrendben vannak, majd a következő számjegyek csökkenő sorrendben vannak a szám végéig.

**Példa** „*hegy-völgy*” számokra: **24521, 18942, 16432**.

1. Olvasd be a billentyűzetről egy **n** és egy **m** elemű, **a** illetve **b** egydimenziós tömb elemeit. Mindkét tömb esetén az elemek növekvő sorrendben vannak. A két tömb elemeiből építs egy harmadik egydimenziós tömböt, optimális módon úgy, hogy a harmadik tömb tartalmazza a két eredeti tömb összes elemét, szintén növekvő sorrendben. A kapott tömb elemeit írasd ki az ***ordonat.txt*** állományba.

**Példa:** ha a beolvasott számok:

**4**

**1 2 3 4**

**5**

**1 3 5 7 9**

Az ***ordonat.txt*** állomány tartalma: **1 1 2 3 3 4 5 7 9**

1. Olvasd be a billentyűzetről egy **n** elemű **a** és egy **m** elemű **b** egydimenziós tömb elemeit. Mindkét tömb esetén az elemek növekvő sorrendben vannak. Ha a két tömb halmaznak tekinthető, határozd meg az egyesített halmazt, illetve a halmazok különbségét. Ha valamelyik tömb nem tekinthető halmaznak (az elemei nem különbözőek), akkor írasd ki a ”**X-nu e multime**” üzentet, az X az **a** vagy a **b** tömb. Az eredményt a ***multime.txt*** állományba kell kiíratni.

**Példa:** ha a beolvasott számok:

**4**

**1 3 5 6**

**5**

**1 3 5 7 9**

Az ***multime.txt*** állomány tartalma:

**1 3 5 6 7 9** (az egyesített halmaz)

**6** (a halmazok különbsége)

1. Az ***interval.txt*** szöveges állomány első sora tartalmazza az n (1≤n≤1000), nullától különböző természetes számot, a következő n sor mindenike pedig az a és b (1≤a≤b≤32000) két egész számot. A számpárok egy-egy [a, b] zárt intervallumot jelölnek. Írj programot, amely kiolvassa a számokat az állományból és meghatározza azt az intervallumot, amelyik a legtöbb egész számot tartalmazza és kiírja a képernyőre az intervallum végpontjait, szóközzel elválasztva. Ha több ilyen intervallum van, akkor annak az intervallumnak a végpontjait kell kiíratni, amelynek a jobboldali végpontja a legkisebb.

**Példa:**

**4**

**17 24**

**-2 3**

**9 15**

**8 15**

A képernyőre kiírt számok: **8 15.** A [8,15] és [17,24] intervallumok azonos számú egész számot tartalmaznak, de a 8 a kisebb.

1. A ***cifre.txt*** szöveges állomány első sora tartalmazza az **n** (0<n<1000), természetes számot, a második sora pedig **n** darab, egyenként legtöbb 9 jegyű természetes számot. Írj programot, amely kiolvassa a számokat az állományból és kiírja a képernyőre, szóközzel elválasztva azokat a háromjegyű számokat, amelyek különböző számjegyeket tartalmaznak.

**Példa**: ha a ***cifre.txt*** állomány tartalma az alábbi:

**7**

**249 511 4329 2 4313 243 3562**

akkor a képernyőre kiírt számok **249 243**

1. A ***vector.in*** állományból kiolvassuk egy egydimenziós tömb elemeit. Az elemek mindenike legalább kétjegyű, pozitív egész szám. Határozd meg a tömb elemei közül azt a leghosszabb, prímeket tartalmazó számsort, amelyek fordítottjai szintén prímek.

Példa: ha a ***vector.in*** állomány tartalma az alábbi:

**9**

**11 971 44 19 181 751 347 33 929**

akkor a kért részsorozat a következő: **181 751 347**

1. Adott, egy pozitív, egész számokat tartalmazó egydimenziós tömb, amelynek elemeit a ***vector.in*** állományból kell kiolvasni. Írasd ki a leghosszabb tükörszámsort.

Példa: ha a ***vector.in*** állomány tartalma az alábbi:

**15**

**1 12 31 12 1 4 27 13 9 26 9 13 27 4 131**

akkor a kért részsorozat a következő: **4 27 13 9 26 9 13 27 4**

1. Olvass be a billentyűzetről egy **n** természetes számot és egy **n** elemű, egydimenziós tömb elemeit. A tömb elemei 1 és 10 közötti egész számok. Határozd meg egy egydimenziós tömbnek az elemeit úgy, hogy a tömb i. pozícióján lévő elem az eredeti tömbben található i értékű elemek előfordulási gyakoriságát jelölje. Az eredményt az ***aparitii.out*** állományba kell kiírni.

Példa: ha **n=9, v=(1, 5, 2, 1, 5, 7, 2, 1, 5)** a kapott tömb **w=(3, 3, 2, 3, 3, 1, 2, 3, 3)**.

**Kétdimenziós tömbök:**

1. Olvass be a billentyűzetről egy n természetes számot (n<=10) majd egy n\*n elemű, egész számokat tartalmazó, egydimenziós tömb elemeit. Tölts fel körkörösen egy kétdimenziós tömböt, az egydimenziós tömb elemeivel, a beolvasás sorrendjében. A kapott kétdimenziós tömböt a ***matrice.txt*** állományba kell kiírni.

**Példa:** **n=4**, az egydimenziós tömb elemei: **1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16.**

A kapott kétdimenziós tömb:

**1 2 3 4**

**12 13 14 5**

**11 16 15 6**

**10 9 8 7**

1. A ***vecini.txt*** állományból olvass ki egy n számot, majd egy négyzetes, egész számokat tartalmazó kétdimenziós tömb elemeit. Írasd a képernyőre, külön-külön sorba, azoknak az elemeknek a koordinátáit, amelyek összes szomszédja (mind a 8 irányban) páratlan szám.

**Példa: n=4,** és amátrix elemei:

**1 1 3 4**

**5 6 7 8**

**9 9 11 12**

**13 14 15 16**

**A kiírt eredmény: 2 2**

1. Írj programot, amely billentyűzetről beolvas egy n természetes számot (n≤50), majd feltölt a memóriában egy n sorból és n oszlopból álló kétdimenziós tömböt, amelynek elemei a billentyűzetről beolvasott egész számok. A program kiírja a ***minim.txt*** állományba, sorrendben a kétdimenziós tömb minden oszlopának legkisebb elemét. A kiírt számokat szóközzel kell elválasztani.

**Példa: n=4, és a mátrix elemei:**

**1 12 13 45**

**5 16 7 18**

**9 10 11 12**

**13 14 15 16**

**A *minim.txt* tartalma: 1 10 7 12**

1. Az ***maxim.txt*** állomány első sora tartalmaz két, *n* és *m*, természetes számot. A második sortól kezdődően az *n\*m* elemű kétdimenziós tömb elemeit találjuk az állományban. Határozd meg a kétdimenziós tömb első, legnagyobb elemét. Írasd a képernyőre a kapott elemet, a koordinátáival együtt. Töröld a tömbből azt a sort és oszlopot, ahol a legnagyobb elemet találtad. A törlés elvégzése után írasd ki a kétdimenziós tömböt.

**Példa:** ha a ***maxim.txt*** tartalma az alábbi:

**4 5**

**1 12 13 45 10**

**5 16 7 18 49**

**9 50 11 12 25**

**13 14 15 16 34**

**A kiírt értékek:**

**50 3 2**

**1 13 45 10**

**5 7 18 49**

**13 15 16 34**

1. A ***matrice***.***in*** állomány első sorából olvass ki két, *n* és *m* értéket, amelyek egy kétdimenziós tömb sor és oszlopszámát jelölik, majd *n\*m* egész értéket. Forgasd el a kétdimenziós tömböt trigonometriai irányban 90 fokkal, majd írasd ki az új kétdimenziós tömböt.

**Példa:** ha a ***maxim.txt*** tartalma az alábbi:

**4 5**

**1 12 13 45 10**

**5 16 7 18 49**

**9 50 11 12 25**

**13 14 15 16 34**

**A kiírt értékek:**

|  |  |  |  |
| --- | --- | --- | --- |
| **10** | **49** | **25** | **34** |
| **45** | **18** | **12** | **16** |
| **13** | **7** | **11** | **15** |
| **12** | **16** | **50** | **14** |
| **1** | **5** | **9** | **13** |

1. Beolvasunk a billentyűzetről egy n páratlan természetes számot. Építs fel és írass ki a ***matrice.txt*** állományba egy, az alábbi formájú kétdimenziós tömböt:

|  |  |  |
| --- | --- | --- |
| N=3 | N=5 | N=7 |
| **1 1 1**  **1 3 1**  **1 1 1** | **1 1 1 1 1**  **1 3 3 3 1**  **1 3 5 3 1**  **1 3 3 3 1**  **1 1 1 1 1** | **1 1 1 1 1 1 1**  **1 3 3 3 3 3 1**  **1 3 5 5 5 3 1**  **1 3 5 7 5 3 1**  **1 3 5 5 5 3 1**  **1 3 3 3 3 3 1**  **1 1 1 1 1 1 1** |

1. Írj programot, amely beolvas a billentyűzetről egy nullától különböző természetes számot. Legyen x a beolvasott szám számjegyeinek a száma. Építs fel a memóriában, majd írass ki a képernyőre egy x soros és x oszlopos kétdimenziós tömböt ahol: az x-ik oszlop elemei az adott szám az egyesek helyén található számjegyét tartalmazzák, az (x-1)-ik oszlop elemei az adott szám tízesek helyén található számjegyét tartalmazzák, … , az első oszlop elemei az adott szám első számjegyét fogják tartalmazni, az alábbi példáknak megfelelően.

|  |  |
| --- | --- |
| N=123 | N=10038 |
| **1 2 3**  **1 2 3**  **1 2 3** | **1 0 0 3 8**  **1 0 0 3 8**  **1 0 0 3 8**  **1 0 0 3 8**  **1 0 0 3 8** |

1. A ***matrice***.***in*** állomány első sorából olvass ki két, n és m értéket, amelyek egy kétdimenziós tömb sor és oszlopszámát jelölik, majd n\*megész értéket. Írasd ki a kétdimenziós tömb „nyeregpontjait”. „Nyeregpont”-nak nevezzük a kétdimenziós tömb azon elemét, amely a sorában legnagyobb és az oszlopában legkisebb elem, vagy fordítva.

**Példa: n=4 m=3**

**1 6 3**

**5 8 2**

**4 7 2**

**5 8 1**

**Az eredmény: 1 2 6 (magyarázat: a[1][2]=6, és ez maximális az 1-es soron és minimális a 2-es oszlopon)**

1. A ***matrice***.***in*** állomány első sora tartalmaz két, *n* és *m* értéket, amelyek egy kétdimenziós tömb sor és oszlopszámát jelölik, majd *n\*m* egész értéket. Határozd meg a kétdimenziós tömb keretén található legnagyobb értéket és ennek előfordulási gyakoriságát. *Keretnek* nevezzük az kétdimenziós tömb első sora, utolsó oszlopa, utolsó sora és első oszlopa elemeiből alkotott vázat.

**Példa:** ha a ***matrice.in*** tartalma az alábbi:

**4 5**

|  |  |  |  |
| --- | --- | --- | --- |
| **10** | **49** | **25** | **34** |
| **49** | **18** | **12** | **16** |
| **13** | **7** | **11** | **15** |
| **12** | **16** | **50** | **14** |
| **1** | **5** | **9** | **49** |

**A kiírt értékek:**

**49 3**

1. Adott egy A négyzetes kétdimenziós tömb, amelynek n sora és n oszlopa van, elemei pedig pozitív értékek. Írj programot, amely beolvassa a kétdimenziós tömb elemeit és helyettesíti a főátlón illetve a mellékátlón található mindenik értéket, a megfelelő sor elemeinek az összegével, majd az eredmény kétdimenziós tömb elemeit kiírja a ***rez.txt*** állományba (a kétdimenziós tömb minden sorának elemeit külön sorba).

**Példa: n=4**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3 | 6 | 1 | 2 |  | 12 | 6 | 1 | 12 |
| 1 | 9 | 4 | 7 |  | 1 | 21 | 21 | 7 |
| 4 | 2 | 6 | 3 | A kapott kétdimenziós tömb: | 4 | 15 | 15 | 3 |
| 2 | 5 | 1 | 8 |  | 16 | 5 | 1 | 16 |

**Karakterláncok:**

1. A ***cuvinte.txt*** állományból olvass ki szavakat melyek hossza legtöbb 20 karakter, a szavak az állomány különböző soraiban vannak. A szavakat írasd a képernyőre növekvő sorrendben, a szavakban előforduló magánhangzók számának függvényében,

**Példa:**

|  |  |
| --- | --- |
| ***cuvinte.txt*** | ***A képernyő tartalma*** |
| **Atestat**  **informatica**  **Bac**  **programator**  **Pascal**  **Cpp**  **Competente** | **Cpp**  **Bac**  **Pascal**  **Atestat**  **Competente**  **programator**  **informatica** |

1. Írj programot, amely beolvas a billentyűzetről egy legtöbb 20 betűs, az angol ábécé kisbetűiből álló szót és kiírja a képernyőre a beolvasott szó összes „klónját” (*s2 karakterlánc az s1 karakterlánc „klónja”, ha s1 karakterláncból törölve egy magánhangzó összes előfordulását s2 karakterláncot kapod*) külön-külön sorba.

**Példa:** az ***informatica*** szó esetén kiírja, nem feltétlenül ebben a sorrendben, a mellékelt „klónokat”: ***nformatca, infrmatica, informtic***.

1. Írj programot, amely beolvas a billentyűzetről két, a c1 és c2 karaktereket és egy legtöbb 250 karakterből álló szöveget (szóközök és az angol abc különböző betűi), amelyet úgy módosít, hogy helyettesíti a c1 karakter minden előfordulását a c2 karakterrel, illetve a c2 karakter minden előfordulását a c1 karakterrel. A program kiírja a képernyő különböző soraiba az eredeti illetve a módosított szöveget.

**Példa:** ha **c1=a és c2=e,** s a szöveg: **“ana are mere”**

Akkor a kiírt eredmény**:**

**ana are mere**

**ene era mara**

1. Olvass be a billentyűzetről egy szöveget. Minden kis magánhangzó után szúrd be a megfelelő nagy magánhangzót. A kapott karakterláncot írasd a ***vocale.txt*** állományba.

**Példa**: az ”**informatica**” átalakul ”**iInfoOrmaAtiIcaA**”.

1. Olvass ki a ***sufix\_prefix.txt*** állomány különböző sorából 2, legtöbb 50 karakter hosszúságú szöveget. Írasd ki, kezdve az első karaktertől, az első szó minden előtagját, majd a második szöveg minden utótagját, kezdve az eredeti szöveggel.

**Példa:** ha a bemeneti állomány tartalma a ”**mate**” és ”**info**” karakterláncok, akkor a képernyő tartalma az alábbi lesz:

**e**

**te**

**ate**

**mate**

**info**

**inf**

**in**

**i**

1. Egy legtöbb 255 karakterből álló karakterlánc, szavakat tartalmaz, egy vagy több szóközzel elválasztva. A szavak az angol ábécé kisbetűiből állnak. Írj programot, amely beolvas egy ilyen karakterláncot, minden szó első és utolsó betűjét nagybetűre változtatja és kiírja a módosított formáját a ***litere.txt*** állományba.

**Példa:** az ”**atestat la informatica**” módosított formája ”**AtestaT LA InformaticA**”.

1. A ***text.in*** állományból olvass ki egy *n* természetes értéket, majd a következő *n* sor mindenikéből egy-egy szót. Számold meg és írasd ki, hogy a kiolvasott szavak közül hány kezdődik és végződik magánhangzóval.

**Példa:**

**6**

**ana**

**are**

**mere**

**scumpe**

**si**

**ulei**

**A kiírt eredmény: 3 (**az **ana, are, ulei** szavakról van szó**)**

1. A ***text.in*** állományból olvass ki egy *n* természetes értéket, majd a következő *n* sor mindenikéből egy-egy szót. Számold meg és írasd ki, hogy a kiolvasott szavak közül hány tartalmaz ugyanannyi magánhangzót, mint mássalhangzót.

**Példa:**

**6**

**ana**

**are**

**mere**

**scumpe**

**si**

**ulei**

**A kiírt eredmény: 2 (**a **mere, si** szavakról van szó**)**

1. Adott egy legtöbb 100 karaktert tartalmazó karakterlánc, amely az angol ábécé betűiből alkotott szavakat tartalmaz, szóközzel elválasztva. Építs egy új karakterláncot, amelyben a magánhangzók annyiszor többszöröződjenek, mint amennyi a sorszámuk az eredeti karakterláncban.

**Példa:**

Az adott karakterlánc: **Cerul este alb.**

A felépített karakterlánc: **Ceeruuuul eeeeeeesteeeeeeeeee aaaaaaaaaaaalb.**

1. A ***in.txt*** állomány egy legtöbb 200 karakter hosszúságú szöveget tartalmaz. Írj programot, amely kódolja az adott szöveget a következő képpen: minden magánhangzó után beszúrja a „p” betűt és a magánhangzót. A kódolt szöveget kiírja a képernyőre.

**Példa:** ha az ***in.txt*** állomány tartalma a „**Mere pere banana**”,

akkor a képernyőre kiírt szöveg: „**Meperepe peperepe bapanapanapa**”

1. Olvass be egy számjegy-betű, számjegy-betű, …stb. alakú karakterláncot. Építsd fel majd írasd ki a ***sir.out*** állományba a beolvasott karakterlánc dekódolt alakját, amelyben minden betű annyiszor fordul elő, mint az őt megelőző számjegy.

**Példa:** ha a bemeneti karakterlánc a **2a4b5c**, a ***sir.out*** kimeneti állományba kiírt karakterlánc a következő **aabbbbccccc**