Alkalmazott modul III
1. feladatcsoport

Közös követelmények:

- A program legyen informatív, a bemenetet várja billentyűzetről, vagy fájlból, és az eredményt írja a konzolra. Fájlból történő bevitel esetén a program kérdezze meg a fájlnevet, és jelezze, ha nem olvasható, vagy tartalma nem megfelelő. Konzolról történő adatbeolvasás esetén is ellenőrizni kell az adatok helyességét. Az ellenőrzéshez használjunk kivételkezelést.

- Az adott feladatot valósítsuk meg alprogramok segítségével, a beolvasásért, kiírásért, valamint a tényleges feladatmegoldásért feleljenek külön alprogramok. Programozási tétel használata esetén minden tétel kerüljön külön alprogramba. Az alprogramok a kommunikációhoz használjanak paraméterátadást és visszatérési értéket.

- Az 1-8. feladatokat programozási tételek segítségével kell megoldani. A programozási tételeket a feladatnak megfelelően kell kiválasztani. Adatok tárolására tetszőleges adatszerkezet használható.

- A 9-16. feladatokat adatszerkezet segítségével kell megvalósítani. A megoldáshoz használt adatszerkezet lehet verem vagy sor. Emellett az adatok tárolására használható egyéb adatszerkezet is (pl. tömb, lista).

Feladatok:

1. Egy kutya kiállításon *n* kategóriában *m* kutya vesz részt. Minden kutya minden kategóriában egy *0* és *10* közötti pontszámot kap. Az első két adat *n* és *m* értéke, ezt követik az értékek kategóriánként. Készítsünk programot, amely megállapítja, hány kategóriát nyert az abszolút győztes kutya, akinek összpontszáma a legnagyobb.

   Példa:
   INPUT: **3 4 10 4 7 3 8 6 9 8 10 5 8 10**
   OUTPUT: **Az abszolút győztes kutya 2 kategóriában nyert.**

2. Madarak életének kutatásával foglalkozó szakemberek *n* különböző településen *m* különböző madárfaj előfordulását tanulmányozzák. Egy adott időszakban megszámolták, hogy az egyes településen egy madárfajnak hány egyedével találkoztak. Az első két adat *n* és *m* értéke, ezt követik az értékek településenként. Készítsünk programot, amely megállapítja, hány településen fordult elő mindegyik madárfaj.

   Példa:
   INPUT: **2 5 14 8 42 3 32 8 0 13 0 67**
   OUTPUT: **1 településen fordult elő mindegyik madárfaj.**

3. Készítsünk programot, amely egy egész számokat tartalmazó négyzetes mátrixról megállapítja, van-e olyan oszlopa, ahol a főátló alatti elemek mind nullák. Az első adat a mátrix mérete, majd ezt követően sorfolytonosan következnek az értékek.

   Példa:
   INPUT: **4 8 0 3 -56 -9 1 14 0 43 0 -31 -5 13 9 -53 -12**
   OUTPUT: **A mátrixnak van olyan oszlopa, ahol teljesül a felvétel.**

4. Készítsünk programot, amely eldönti, hogy egy egész számokat tartalmazó mátrixban minden sorban van-e legalább egy prímszám. A program először megkapja a mátrix sorainak, majd oszlopainak számát, és ezt követik sorfolytonosan az értékek.

   Példa:
   INPUT: **4 3 8 0 3 -9 1 10 43 0 -31 13 9 -12**
   OUTPUT: **A mátrixnak nincs minden sorában prímszám.**

5. A tornaórán névsor szerint sorba állítottuk a diákokat, és megkérdeztük a testmagasságukat. Készítsünk programot, amely a magassági adatokat megkapva megállapítja, hányadik diákot előzte meg a legtöbb nála magasabb.

   Példa:
   INPUT: **187 175 159 182 167 174 172 185**
   OUTPUT: **A(z) 7. diákot előzte meg a legtöbb nála magasabb.**

6. Egymást követő napokon délben megmértük a levegő hőmérsékletét. Készítsünk programot, amely megállítja melyik érték fordult elő leggyakrabban!

   Példa:
   INPUT: **27.3 31 29.1 27.3 27.8 27.2 33.2 31 28.6 27.3**
   OUTPUT: **A(z) 27.3 fokos hőmérséklet fordult elő legtöbbször.**

7. Feljegyeztük, hogy egymás követő hétvégeken hány Forintot nyertünk vagy veszítettünk a lóversenyen. Készítsünk programot, amely megállapítja, mikor volt a legnagyobb a nyereségünk összege.

   Példa:
   INPUT: **6400 -2000 -4300 8200 1000 -3400 600 -900**
   OUTPUT: **A(z) 5. héten volt a legnagyobb a nyereségünk.**

8. Egy határállomáson feljegyezték az átlépő utasok útlevélszámát. Készítsünk programot, amely megállapítja, melyik útlevélszámú utas fordult meg leghamarabb másodszor a határon.

   Példa:
   INPUT: **CJ8345634 KB2590621 ZD1206851 KB2590621 KN4873965 ZD1206851**
   OUTPUT: **A KB2590621 útlevélszámú utas ismétlődik leghamarabb.**

9. Készítsünk programot, amely egy megadott szöveg szavainak sorrendjét megfordítja. A bemenet egy szóközzel elválasztott szavakból álló szöveg, a kimenetben pedig a szöveg szavait kell kiírni fordított sorrendben. A megoldáshoz használjunk karaktereket tároló verem adatszerkezetet.

   Példa:
   INPUT: **`szépen süt a nap`**
   OUTPUT: **`nap a süt szépen`**

10. Készítsünk programot, amely egy **`(`**, **`)`** jeleket tartalmazó kifejezésről eldönti, hogy helyes zárójelezés-e, és kiírja az összetartozó zárójelpárok sorszámát. Egy zárójelezés akkor helyes, ha minden nyitott zárójelet lezárunk, és egyetlen ponton sem zárunk be több zárójelet, mint amennyit addig megnyitottunk.

    Példák:
    INPUT: **`()(())`**
    OUTPUT: **`helyes, zárójelpárok: 1:2 4:5 3:6`**

    INPUT: **`(()))(()`**
    OUTPUT: **`nem helyes`**

11. Készítsünk programot, amely egy `(`, `)`, `[`, `]`, `{`, `}` jeleket tartalmazó kifejezésről eldönti, hogy helyes zárójelezés-e. Egy ilyen zárójelezést akkor tekintünk helyesnek, ha minden nyitó zárójelet a megfelelő párja zár le, továbbá ha a zárójelpárok a matematikai szabályoknak megfelelően vannak egymásba ágyazva, tehát szögletes zárójelen belül kapcsos, valamint kerek zárójelen belül szögletes és kapcsos zárójel nem állhat.

    Példák:
    INPUT: **`[()](){{()}[]}`**
    OUTPUT: **`helyes`**

    INPUT: **`([)]`**
    OUTPUT: **`nem helyes`**

    INPUT: **`{([])]}`**
    OUTPUT: **`nem helyes`**

12. Készítsünk programot, amely kiírja egy megadott aritmetikai kifejezés lengyelformáját. Az inputkifejezés operandusokat, műveleti jeleket és kerek zárójeleket tartalmazhat. Egy operandus egy egyetlen betűből álló változónév vagy egy egyetlen számjegyből álló természetes szám lehet. A műveleteket a **`+`**, **`-`**, **`*`**, **`/`**, **`^`**, **`=`** jelekkel jelöljük, amelyeket a szokásos precedenciákkal és zárójeleződési (asszociativitási) szabályokkal kell értelmezni. Megengedett a többszörös értékadás is (pl. **`a=b=0`**)!

    Példák:
    INPUT: **`2*(a+3)`**
    OUTPUT: **`2a3+*`**

INPUT: **z=3+2*x-5/2**
OUTPUT: **z32x*+52/-=**

INPUT: **x=y=x+(b-1)^2^b+(a-4-b)*(2*(3+x)+c)**
OUTPUT: **xyxb1-2b^^+a4-b-23x+*c+*+==**

13. Készítsünk programot, amely kiírja a Pascal-háromszög első *n* sorát. A programban szorzás és osztás műveleteket nem használhatunk.

    Példa:
    INPUT: **5**
    OUTPUT:
    ```
    1
    1 1
    1 2 1
    1 3 3 1
    1 4 6 4 1
    ```

14. Készítsünk programot, amely kiír a képernyőre *n* sort úgy, hogy az *k*-adik sorban 2^k-szor szerepel a *k* szám. A sorok számozását nullától kezdjük. A programban szorzás és osztás műveleteket nem használhatunk!

    Példa:
    INPUT: **5**
    OUTPUT:
    ```
    0
    11
    2222
    33333333
    4444444444444444
    ```

15. Adott két szöveg, **x** és **y**. Készítsünk programot, amely eldönti, hogy az **y** szöveg az **x** szöveg ciklikus ismétlésével előáll-e. Pontosabban, döntsük el, hogy az **y** szöveg **vxxx...xw = vxkw** alakú-e, ahol **v** az **x** egy szuffixe, w pedig az **x** egy prefixe. v és w lehet üres is, és az x szó teljes ismétléseinek száma tetszőleges **k≥0** érték lehet. Tegyük fel, hogy az **x** szöveg minden karaktere különböző.

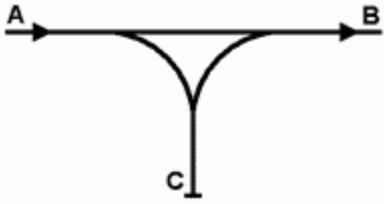
    Példák:
    INPUT: **abc abcab**
    OUTPUT: **igen**

    INPUT: **abcd dabcdabcdab**
    OUTPUT: **igen**

    INPUT: **abc bca**
    OUTPUT: **igen**

    INPUT: **abc bcabcb**
    OUTPUT: **nem**

16. Adott egy vasúti rendezőpálya, amelynek szerkezete az alábbi ábrán látható.



Egy szerelvény érkezik az *A* vágányra, amelyet megfelelően átrendezve kell továbbküldeni a *B* vágányra. Az átrendezéshez felhasználhatjuk a *C* vonalat, amely egy "zsákutca", viszont feltehető, hogy akár az összes vagon egyszerre elfér rajta. Az átrendezés során az *A* vágányra érkező szerelvény vagonjait sorban egymás után át kell tolnunk vagy a *B* vagy *C* vágányra, de közben a *C* vonalról is mozgathatunk vagonokat a *B*-re. A *C* vágányról az ábrának megfelelően mindig a legutoljára odakerült vagont tolhatjuk át a *B*-re. Az átrendezés végén minden vagonnak meg kell érkeznie a *B* vágányra. Készítsünk programot, amely eldönti, hogy egy adott szerelvény egy megadott sorrendben átrendezhető-e.

Példák:
INPUT: **`abcdef ebadcf`**
OUTPUT: **`igen`**

INPUT: **`abcd cbad`**
OUTPUT: **`igen`**

INPUT: **`abcd bcad`**
OUTPUT: **`nem`**