## Feladat

*Valósítsuk meg a diagonális mátrixtípust (amelynek négyzetes mátrixai csak a főátlójukban tartalmazhatnak nullától különböző számot)! Ilyenkor elegendő csak a főátló elemeit reprezentálni egy sorozatban. Implementáljuk a mátrix i-edik sorának j-edik elemét megváltoztató illetve azt lekérdező műveletet, valamint az összeadás és szorzás műveleteket! Ne feledkezzünk meg a megfelelő beolvasó és kiíró műveletekről sem!*

## Diagonális mátrix típus

A feladat lényege egy felhasználói típusnak a diagonális mátrix típusnak a megvalósítása.

### *Típusérték-halmaz*[1]

Olyan számokat (ebben az esetben egész számokat: $\mathbb{Z}$) tartalmazó $n\times n$-es ($n\in\mathbb{N}$) négyzetes mátrixokkal akarunk dolgozni, amelyek csak a főátlójukban tartalmazhatnak nullától különböző elemeket. Az $n\in\mathbb{N}$ ennek a típusnak egy paramétere, amely a típusérték-halmaz mátrixainak méretét határozza meg.

Formálisan: $Diag(n) = \{\ a\in \mathbb{Z}^{n\times n}\ \vert\ \forall i,j\in[1..n]: i\neq j \rightarrow a[i,j]=0\ \}$

### *Típus-műveletek*[2]

*1. Lekérdezés*

A mátrix $i$-edik sorának $j$-edik pozícióján ($i,j\in[1..n]$) álló érték kiolvasása: $e:=a[i,j]$.

Formálisan:

$$A : \underset{a}{Diag(n)} \times \underset{i}{\mathbb{Z}} \times \underset{j}{\mathbb{Z}} \times \underset{e}{\mathbb{Z}}$$

$$Q : (\ a=a' \wedge\ i=i' \wedge\ j=j' \wedge i,j\in[1..n]\ )$$

$$R : (\ Q \wedge e=a[i,j]\ )$$

Megjegyezzük, hogy ez a művelet csak $i=j$ esetén igényel tényleges tevékenységet, hiszen egyébként a visszaadott elem nulla.

*2. Felülírás*

A mátrix $i$-edik sorának $j$-edik pozíciójára ($i,j\in[1..n]$) új érték beírása: $a[i,j]:=e$. A főátlón kívüli elemeket nem szabad felülírni, azaz $i=j$.

Formálisan:

$$A : \underset{a}{Diag(n)} \times \underset{i}{\mathbb{Z}} \times \underset{j}{\mathbb{Z}} \times \underset{e}{\mathbb{Z}}$$

$$Q : (\ e=e' \wedge a=a' \wedge\ i=i' \wedge\ j=j' \wedge i,j\in[1..n] \wedge\ i=j\ )$$

$$R : (e=e' \wedge\ i=i' \wedge\ j=j' \wedge a[i,j]=e \wedge \forall k,l\in[1..n]: (k\neq i \vee l\neq j)\rightarrow a[k,l]=a'[k,l]\ )$$

Megjegyezzük, hogy ez a művelet csak $i=j$ esetén igényel tényleges tevékenységet; $i\neq j$ esetén hibás, amennyiben egy nemnulla értéket akarunk a mátrixba tenni.

---

[1] A típusérték-halmazt kétféleképpen is le lehet írni: szövegesen és formálisan. Elég csak az egyik formát használni.

[2] A típusműveletek leírására is kétféle definíciót használok: egy informálist és egy formálist. Elég csak az egyik formát használni.

*3. Összeadás*
Két mátrix összeadása: $c:=a+b$. Az összeadásban szereplő mátrixok azonos méretűek.

Formálisan: $A : \underset{a}{Diag(n)} \times \underset{b}{Diag(n)} \times \underset{c}{Diag(n)}$

$Q : ( a=a' \wedge b=b')$

$R = ( Q \wedge \forall i,j\in[1..n]: c[i,j]= a[i,j] + b[i,j] )$

Diagonális mátrixok esetén a fenti művelet jóval egyszerűbben is megfogalmazható:

$\forall i\in[1..n]: c[i,i]= a[i,i] + b[i,i]$ és $\forall i,j\in[1..n]: i\neq j \rightarrow c[i,j]=0.$

*4. Szorzás*
Két mátrix összeadása: $c:=a*b$. Az összeadásban szereplő mátrixok azonos méretűek.

Formálisan: $A : \underset{a}{Diag(n)} \times \underset{b}{Diag(n)} \times \underset{c}{Diag(n)}$

$Q : ( a=a' \wedge b=b')$

$R : ( Q \wedge \forall i,j\in[1..n]: c[i,j]= \Sigma_{k=1..n}\, a[i,k] * b[k,j])$

Diagonális mátrixok esetén a fenti művelet jóval egyszerűbben is megfogalmazható:

$\forall i\in[1..n]: c[i,i]= a[i,i]*b[i,i]$ és $\forall i,j\in[1..n]: i\neq j \rightarrow c[i,j]=0.$

## ***Reprezentáció***

Egy $n\times n$-es diagonális mátrixnak csak a főátlóját kell ábrázolni.

$$\begin{matrix} a_{11} & 0 & 0 & \dots & 0 \\ 0 & a_{22} & 0 & \dots & 0 \\ 0 & 0 & a_{33} & \dots & 0 \\ 0 & 0 & 0 & \dots & a_{nn} \end{matrix} \quad \leftrightarrow \quad a_{11} \quad a_{22} \quad a_{33} \quad a_{nn}$$

Ehhez egy $0$-tól $n-1$-ig indexelt egydimenziós tömbre ($v$) van szükségünk. Ennek segítségével a diagonális mátrix bármelyik elemét meghatározhatjuk az alábbi képlet alapján:

## ***Implementáció***[3]

*1. Lekérdezés*

A $v$ tömbbel ábrázolt $a$ mátrix $i$-edik sorának $j$-edik elemét visszaadó $e:=a[i,j]$ értékadás az alábbi programmal implementálható feltéve, hogy $1 \le i \le n$, ahol $n$ a mátrix mérete:

| $i=j$ | |
|---|---|
| $e:=v[i-1]$ | $e:=0$ |

*2. Felülírás*

A $v$ tömbbel ábrázolt $a$ mátrix $i$-edik sorának $j$-edik elemét megváltoztató $a[i,j]:=e$ értékadás az alábbi programmal implementálható feltéve, hogy $1 \le i \le n$, ahol $n$ a mátrix mérete.

| $i=j$ | |
|---|---|
| $v[i-1]:=e$ | $SKIP$ |

*3. Összeadás*

A $v$ tömbbel ábrázolt $a$ mátrix és a $t$ tömbbel ábrázolt $b$ mátrix összege az $u$ tömbbel ábrázolt $c$ mátrixba kerül, ha az alábbi programot végrehajtjuk. A végrehajtás előtt ellenőrizni kell, hogy mindhárom mátrix, pontosabban az őket reprezentáló tömb azonos méretű-e.

$$\forall i \in [0..n-1]: u[i]:= v[i] + t[i]$$

*4. Szorzás*

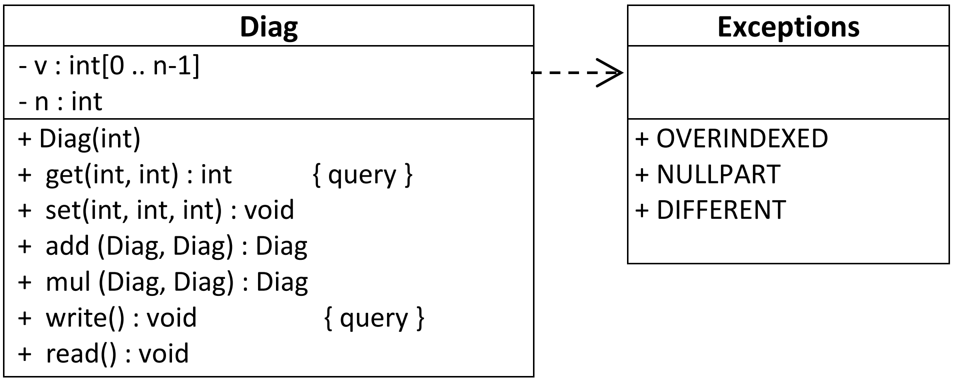
A $v$ tömbbel ábrázolt $a$ mátrix és a $t$ tömbbel ábrázolt $b$ mátrix szorzata az $u$ tömbbel ábrázolt $c$ mátrixba kerül, ha az alábbi programot végrehajtjuk. A végrehajtás előtt ellenőrizni kell, hogy mindhárom mátrix, pontosabban az őket reprezentáló tömb azonos méretű-e.

$$\forall i \in [0..n-1]: u[i]:= v[i] * t[i]$$

[3] A műveletek implementálásához mindig egy programot kell megadni (de nem feltétlenül struktogram alakban).

### *Osztály*

A diagonális mátrixok típusát egy osztály segítségével valósítjuk. A konstruktoron keresztül állítható be a mátrix mérete. A műveleteknél majd ellenőrizni kell, hogy csak azonos méretű mátrixokat lehet összeadni, szorozni, egyébként dobjunk kivételt.




A főátlóbeli elemek tárolására szolgál tömböt ábrázolhatjuk vector<int>-ként (C, D, és E szakirány), de dinamikusan lefoglalt tömbként is (A és B szakirány). Az utóbbi esetben a konstruktor foglalja le a diagonális elemek számára a dinamikus tömböt, a destruktor végzi a felszabadítást, és szükség lesz az értékadás operátor és a másoló konstruktor felüldefiniálására.

A mátrix adott koordinátájú elemének kiolvasására illetve felülírására a () operátor kétféle felüldefiniálását használjuk. Az összeadás, a szorzás műveleteket külső barát függvényként felüldefiniált változataival valósítjuk meg.

Kiegészítjük még az osztályt a mátrix kiírását és a beolvasását végző metódusokkal, amelyek a kiíró és beolvasó operátorok külső barát függvényként felüldefiniált változataival valósítunk meg.

A hibakezelésre az első esetben három, a második esetben kettő kivételt definiálunk. Az `OVERINDEXED` a helytelenül megadott sor és oszlopindexek esetén váltódik ki a mátrix elemeit lekérdező és felülíró műveletekben. A `NULLPART` kivétel a főátlón kívüli elemek felülírásakor aktivizálódik. A `DIFFERENT` kivétel a különböző méretű mátrixok esetén váltódik ki az értékadás, az összeadás és a szorzás műveletekben.

## Tesztelési terv

### Megvalósított műveletek tesztelése (fekete doboz tesztelés)

1) Különféle méretű mátrixok létrehozása, feltöltése és kiírása.
   a) 0, 1, 2, 5 dimenziójú mátrix
2) Mátrix adott pozíciójú értékének lekérdezése és megváltoztatása.
   a) Diagonálisra eső elem lekérdezése és megváltoztatása
   b) Diagonálison kívüli elem lekérdezése és megváltoztatása
   c) Illegális index megadása, 0 dimenziós mátrix indexelése
3) A másoló konsruktor kipróbálása.
   a) A *b* mátrix létrehozása az *a* mátrix mintájára, majd a két mátrix tartalmának összehasonlítása, majd az egyik mátrix megváltoztatása és a két mátrix tartalmának összehasonlítása.
4) Az értékadás operátor kipróbálása.
   a) A *b=a* értékadás végrehajtása az *a* és *b* mátrixokra (az *a* és *b* mérete azonos illetve különbözik: egyiknek illetve másik a nagyobb), majd a két mátrix tartalmának összehasonlítása, majd az egyik mátrix megváltoztatása és a két mátrix tartalmának összehasonlítása.
   b) A *c=b=a* értékadás végrehajtása az *a*, *b* és *c* mátrixokra (ezek mérete lehet különböző), majd a két mátrix tartalmának összehasonlítása, majd az egyik mátrix megváltoztatása és a mátrixok tartalmának összehasonlítása.
   c) Az *a=a* értékadás végrehajtása az *a* mátrixra, majd az *a* mátrix kiírása.
5) A *c*:=*a*+*b* mátrixösszeadás kipróbálása.
   a) Eltérő méretű mátrixokkal (az *a* és *b* mérete különbözik, a *c* és *a* mérete különbözik)
   b) Kommutatívitás ellenőrzése (a + b == b + a)
   c) Asszociatívitás ellenőrzése (a + b + c == (a + b) + c == a + (b + c))
   d) Null elem vizsgálata (a + 0 == a, ahol 0 a null mátrix)
6) A *c*:=*a***b* mátrixszorzás kipróbálása.
   a) Eltérő méretű mátrixokkal. (az *a* és *b* mérete különbözik, a *c* és *a* mérete különbözik)
   b) Kommutatívitás ellenőrzése (a * b == b * a)
   c) Asszociatívitás ellenőrzése (a * b * c == (a * b) * c == a * (b * c))
   d) Null elem vizsgálata (a * 0 == 0, ahol 0 a null mátrix)
   e) Egység elem vizsgálata (a * 1 == a, ahol 1 az egység mátrix)

Megj: A beolvasó és kiíró operátorok teszteléséhez elég, hogy ezeket a fenti esetek tesztelésénél intenzíven használjuk.

Tesztesetek a kód alapján (fehér doboz tesztelés)

1. Extrém méretű (-1, 0, 1, 1000) mátrix létrehozása.
2. Kivételek generálása és elkapása.