# Feladat

Valósítsa meg a nagyon nagyszámok típusát! Ábrázoljuk a számokat a

számjegyeik láncolt listáival és implementálja az összeadást és a szorzást operátor

túldefiniálással! Ne feledkezzen meg a beolvasó (operátor>>) és kiíró (operátor<<) metódusokról sem! Az összeadás műveletigénye Ơ(m+n), a szorzásé Ơ(m\*n), ahol m és n a két szám számjegyeinek száma.

# Nagy számok típusa

## Tipusérték-halmaz

BigInt∈N

## Tipus-műveletek

+:BigInt×BigInt→BigInt

\*:BigInt×→BigInt

\*:BigInt×BigInt→BigInt

## Összehasonlító műveletek:

==:BigInt×BigInt→Boolean

!=:BigInt×BigInt→Boolean

## Bejáró műveletek:

First: a lista első elemére áll

Next: a lista következő elemére áll

End: jelzi, hogy a lista végére értünk-e

Current: az aktuális értéket adja vissza

## Reprezentáció

A számokat(nem negatív egész) számjegyenként történő ábrázolását kétirányú láncolt lista segítségével(BigInt) valósítjuk meg, ahol az adott számjegynek egy listaelem(Node:struct) felel meg.

BigInt:

* first: lista első elemére mutat (üres lista esetén a null pointer)
* last: lista utolsó elemére mutat (üres lista esetén a null pointer)
* current: bejáráskor a lista aktuális elemére mutat, egyébként nem használjuk

Továbbá tároljuk a listaelemek számát a size változóban.

Node:

|  |  |  |
| --- | --- | --- |
| PREV | VALUE | NEXT |

* PREV: A lista adott elemét megelőző elemre mutat (memóriacímére).
* NEXT: A lista adott elemét követő elemre mutat (memóriacímére).
* VALUE:A lista adott elemének értékét tárolja.

## Implementáció

Az osztály deklarációja:BigInt.h

A nagy számok osztályát és műveleteit egy külön modulban implementáljuk: BigInt.cpp

Az osztály:

* összeadás(+)
* szorzás(\*)
* kiírás(<<)
* beolvasás(>>)
* egyenlő(==)
* nem egyenlő(!=)

műveleteit barát függvényekkel implementáljuk.

## Osztály

|  |
| --- |
| **Node** |
| value:short  next:Node\*  prev:Node\* |

A nagyszámok (BigInt) típust egy osztály segítségével valósítjuk meg.

|  |
| --- |
| **BigInt** |
| -Node:Struct  -int:size  -first:Node\*  -last:Node\*  -current:Node\* |
| +(+)(BigInt,BigInt):BigInt  +(\*)(BigInt,int):BigInt  +(\*)(BigInt,BigInt):BigInt  +(==)(BigInt,BigInt):bool  +(!=)(BigInt,BigInt):bool |

|  |
| --- |
| **Exeptions** |
|  |
| +WRONG\_VALUE  +NOTBIG |

Exeptions:

2 kivételt definiálunk

WRONG\_VALUE:nem szám karakter megadása

NOTBIG:amennyiben negatív számot adunk értékül

A kivételek elkapásáért a tesztkörnyezet (menü) felel.

## Teszkörnyezet:

A létrehozott osztály és annak műveleteit egy menü segítségével teszteljük.

A menüt deklarációja és implementációja a menu.h headerben és menu.cpp állományban található osztály segítségével megvalósítva.

Menü feladata a nagy számok létrehozása és műveleteik elérése, keletkezett kivételek fogása.

A menü példányósítását és futtatását a főprogram végzi.

## Tesztelés:

* Különböző nagyságú létrehozása, beleértve nulla, illetve extrém nagy méreteket.
* Érték felülírásának ellenőrzése számmal illetve más karakterrel.
* Műveletvégzés (+, \*) különböző méretű számokon. Műveltek kipróbálása egység illetve nullelemekkel.
* Kivételek generálása.