A házik beadásához küldjétek el a megoldásokat (.c, .cpp, ha van .h és .hpp file-ok) csatolva a következõ emailcímre: tofihomework+2024info3hu@gmail.com

A lefordított programokat nem kell elküldeni. Ha úgy érzitek túl sok file-t küldenétek (5+) akkor bedobhatjátok egy zip-be, de nem muszáj.

Halmaz (folytatás)

Ez a feladat az 5. házi folytatása, be lehet küldeni õket egyben vagy csak az 5. házit. Együtt természetesen két házinak számít így 2szer annyi pontot ér.

Írjunk halmaz osztályt, ami fix méretû egész számok halmazt tud reprezentálni.

- Létrehozáskor kelljen megadni a halmaz maximális méretét. Azaz, hogy maximum hány elem fér bele.
- Lehessen hozzáadni új elemet. Ha az adott szám már benne van a halmazban, akkor ne adja hozzá újra és adjon vissza **false**-t. Ha az elem még nincs benne, akkor adja hozzá és adjon vissza **true**-t.
- Lehessen copy konstruktorral lemásolni egy halmazt.
- Oldjuk meg, hogy lehessen hozzáadni halmazhoz egész számot a + mûvelettel. Ez hasonló az elõzõ metódushoz, de itt ne a létezõ halmazhoz adjuk hozzá az elemet, hanem készítsünk új halmazt amiben benne van az az elem is és adjuk vissza ezt az új halmazt.
- Lehessen elkérni a halmaz méretét, azaz, hogy jelenleg hány elemet tartalmaz.
- Legyen **in** metódusa, ami megnézi, hogy a kapott egész szám benne van-e a halmazban vagy sem. (true/false-t adjon vissza.)
- Oldjuk meg, hogy ha két halmazt összeadunk a + mûvelettel, akkor a két halmaz unio-ját kapjuk.
- Oldjuk meg, hogy ha két halmazt kivonunk, akkor a két halmaz különbségét kapjuk. (A - B-ben azok az elemek vannak amik A-ban benne vannak, de B-ben nem.)
- A fenti metódusoknál amikor két halmazon végzünk mûveleteket az eredmény halmaz mérete legyen a nagyobb a két halmaz mérete közül.
- Nem kell figyelnünk arra, hogy biztosan belefér-e az összes elem az eredmény halmazba. (De ha van rá idõdök attól még megoldhatjátok, nem nehéz, ha kifutnánk az elemekbõl, akkor szimplán duplázzuk a tömb méretét és másoljuk át az eddigi elemeket.)
- Lehessen elkérni elemeket az **operator[]**-al. Mindegy milyen sorrendben adja vissza, de a szokásostól eltérõen ne lehessen megváltoztatni az elemek értéket ezen keresztül. (Azaz ne lehessen balértékként használni.)
- Lehessen **cout**-al kiírni halmazt. Így nézzen ki egy halmaz kiírva (sorrend nem számít):

```
(1, 8, 5)
```

Példa main függvény tesztelésre az elején:

```
int main(void) {
  Halmaz A = Halmaz(10);
  A.add(23);
  Halmaz B = A;
  Halmaz C = A + 12;
  Halmaz D = A - 23;
  Halmaz E = A + 12;
  cout << B.in(23) << endl;
  cout << C.in(12) << endl;
  cout << D.in(23) << endl;
  cout << E.hossz() << endl;
  return 0;
}
```

Példa main függvény tesztelésre a végén:

```
int main(void) {
  Halmaz A = Halmaz(100);
  Halmaz B = Halmaz(200);
  for(int i = 0; i < 100; i++) {
    if(i % 2 == 0) { // A-ban párosak
      A.add(i);
    }
    if(i % 3 == 0) { // B-ben 3-al oszthatóak
      B.add(i);
    }
  }
  Halmaz C = A + B;
  Halmaz D = A - B;
  for(int i = 0; i < D.hossz(); i++) {
    cout << D[i] << endl;
  }
  // A[0] = 10; // Ez ne mûködjön!
  cout << C << endl;
  return 0;
}
```

Tipp a lap alján (ha van rá idõtök/kitartásotok, elõször próbáljátok meg nélküle megoldani/végiggondolni).

A feladat nem igazán nehéz, inkább csak hosszú. Szóval ne ijedjünk meg tõle, kezdjünk bele és gyorsan meglesz.

Tároláshoz dinamikusan foglalt tömböt javaslok. A konstruktorban hozzuk létre a megfelelõ méretû tömböt. Továbbá tároljuk a jelenleg tárolt elemek számát is. Ne felejtsük el megírni a destruktort is. Figyeljünk arra, hogy a copy konstruktorban új tömböt kell létrehoznunk, nem elég csak a másolandó halmaz pointerét lemásolni.

Arra kell igazán még vigyázni a metódusokban, hogy mindig jól állítsuk a jelenleg tárolt elemek adattagot. Pl ha hozzáadunk új elemet akkor növeljük, de ha az elem már benne volt a halmazban, akkor ne növeljük (hisz ekkor nem rakjuk bele a tároló tömbbe). Lehet továbbá az **add** vagy + operátort használni amikor az unio-t írjuk, valamint az **in**-t, amikor a halmaz különbséget.

Amikor az **operator[]**-t írjuk, ahhoz, hogy ne lehessen balértékként használni csak annyi a dolgunk, hogy nem int&-t adunk vissza az **operator[]**-al, hanem csak simán int-et.