

Láda pakolás

Egy gyárban egy nap csak azonos súlyú termékeket gyártanak, és ezeket ládába pakolják, de előre nem tudják, hogy aznap hány terméket fognak gyártani, és azt sem, hogy a további napokon hány darab és milyen súlyú termékek lesznek. Így kénytelenek valami heurisztikán¹ alapuló átlagosan hatékony algoritmust használni arra, hogy hogyan csoportosítsák a termékeket. A ládádkban 100 kilónyi termék fér el, és a termékek súlya csak 1 és 100 közé eső egész szám lehet. A cél természetesen az lenne, hogy minél kevesebb ládába pakoljanak, de mivel előre nem tudják a termékeket, így optimális pakolás helyett a következő algoritmust használják:

Egy elemet azonnal be kell rakni valamelyik ládába amint készen van, és később már nem lehet átrakni másik ládába. A választás a legkisebb négyzetek módszerével történik: jelölje $darab(i)$ azon ládák számát, melyekben éppen i kilónyi elem van külön-külön². Ezek után ha jön egy új kész elem, rakjuk abba a ládába, amire a $\sum darab(i) * darab(i)$ $i = 1, \dots, 99$, azaz a ládák gyakoriságának négyzetösszege az új elem berakása után minimális lesz. A képletből látható, hogy a teljesen teli ládák száma: $darab(100)$ nincs benne a négyzetösszegbe, vagyis a teljesen teli ládákat nagyon szeretjük, mert ekkor a négyzetösszeg jelentősen csökken, hiszen egyik tagot sem növeljük amelyik szerepel a szummában, de lehet hogy az egyiket még csökkentjük is, hiszen ha egy olyan ládába rakjuk bele az i súlyú elemet, amelyikben eddig éppen $(100-i)$ súly volt összesen, akkor ezek után a $darab(100-i)$ eggyel csökken. Ugyan a $darab(100)$ eggyel nő, de ez ugye nem szerepel a szummában. Természetesen új üres ládába is pakolhatunk, ha ez adja a minimális négyzetösszeget. De figyelj rá, hogy a $darab(0)$ szintén nincs benne a szummában. A napok száma $[1,10]$, a ládák száma $[1,100000]$, az elemek súlya $[1,100]$ és az egy napon termelt elemek száma az $[1,1000]$ intervallumba esik. A feladat a megadott napi termelések alapján elvégezni a bepakolást, és kiszámolni a végső $darab(i)$ -ket.

Bemenet

A `feladat.be` fájl első sora egy darab n számot tartalmaz, ami a napok számát jelenti, azaz hogy hány további sor lesz a fájlban. Ezután pontosan n db sor következik, mindegyikben az első szám jelenti hogy aznap, hány darab elemet gyártottak, és a második szám tartalmazza, hogy ezeknek mekkora volt egyenként a súlyuk. A két számot szóköz választja el, és a sor végén nincs space, csak az újsor karakter. Természetesen az utolsó nap adatainak a végén is van újsor.

Kimenet

A `kimenet.be` fájl tartalma csak egyetlen sorból állhat. Benne 100 darab egész szám, konkrétan a $darab(i)$ számok $i = 1, \dots, 100$, tehát a maradék üres ládák száma ($darab(0)$) nem érdekes, viszont a teli ládák számát ($darab(100)$) ki kell íratni. A számok között egy-egy szóköz legyen, és az utolsó szám után is csak egy szóköz kell és újsor nem.

¹becslés

²ez lehet, hogy egy db. i kg-os elem, de lehet hogy több darab és különböző fajta, csak az összsúlyuk éppen i .

kimenet

```
0000010000010000010000010000010000010000000000
001000001000001000001000002000002000003000004
0000060000
```

Hivatkozások

[1] Kövesi-Nagy Dániel: *h254153@stud.u-szeged.hu*