

Kijelentés

Egy programozói versenyen x darab első díjat, y darab második díjat és z darab harmadik díjat osztanak ki. A bizottság az $x + y + z$ nyertes között szeretné szétosztani az m darab nyereménytárgyat, melyet a szponzorok felajánlottak és amelyeknek ismert az értéke. Az első díjazottak szigorúan nagyobb értékű jutalmat kell kapjanak, mint a második díjazottak, akik szigorúan nagyobb értékű jutalmat kell kapjanak, mint a harmadik díjazottak. Egy versenyző több tárgyat is kaphat, de minden első díjazott egyenlő értékben kell kapjon nyereményt, akárcsak a második és harmadik díjazottak. Minden tárgyat át kell adni. Számoljuk meg, hogy hányféleképpen lehet kiosztani a díjakat!

Bemenet

A bemenet első sora m értékét tartalmazza. A második sorban található x , y és z értéke szóközzel elválasztva. A harmadik sorban található pontosan m szám, amelyek a díjak értékeit jelölik.

Kimenet

A kimenetre négy számot tartalmazó sorokat kell írni. Az első szám azt jelöli, hogy milyen értékű nyereményt kapnak egyenként az első díjazottak. Az második szám azt jelöli, hogy milyen értékű nyereményt kapnak egyenként a második díjazottak. A harmadik szám azt jelöli, hogy milyen értékű nyereményt kapnak egyenként a harmadik díjazottak. A negyedik szám azt jelöli, hogy hányféleképpen lehet szétosztani a nyereményeket az első három szám által jelölt értékek betartásával.

Megkötések

- $1 \leq m \leq 10$
- $1 \leq x, y, z \leq 10$
- a tárgyak értékei egész számok az $[1, 1000]$ intervallumból
- a kimeneti állomány sorai lexikografikus sorrendben kell legyenek
- két díjazási konfigurációt azonosnak tekintünk, ha csak abban különböznek, hogy két azonos díjat kapó nyertes által kapott tárgyak fel vannak cserélve a két díjazott között
- két azonos értékű tárgyat különbözőnek tekintünk a konfigurációk számolásakor
- ha nincs megoldás, írjunk ki -1-et

Példák

Bemenet	Kimenet
7	12 8 5 1
2 2 1	12 9 3 1
5 8 3 7 12 1 9	

Magyarázat:

A kimeneti állomány első sorában a következő konfigurációt számoljuk: az egyik első díjazott megkapja a 12 értékű tárgyat, a másik a 9 és a 3 értékű tárgyakat. Az egyik második díjazott megkapja a 8 értékű tárgyat, a másik a 7 és az 1 értékű tárgyakat. A harmadik díjazott megkapja az 5 értékű tárgyat.

A kimeneti állomány második sorában a következő konfigurációt számoljuk: az egyik első díjazott megkapja a 12 értékű tárgyat, a másik az 5 és a 7 értékű tárgyakat. Az egyik második díjazott megkapja a 9 értékű tárgyat, a másik a 8 és az 1 értékű tárgyakat. A harmadik díjazott megkapja a 3 értékű tárgyat.

Maximálisan megengedett futási idő: 1 másodperc

Útmutatás: Először generáljuk a díjak lehetséges értékeit (pl. 12, 8, 5). Egy ilyen konfigurációra, generáljuk mind az $x + y + z$ díjazott esetén, hogy melyik tárgyakat kapja meg. Mindegyik díjazott esetén generáljuk az általa kapott tárgyakat, figyelve arra, hogy ne kapjon olyan tárgyat, amit más díjazott már megkapott. Tehát tulajdonképpen a megoldás backtracking a backtrackingben, amit lehet implementálni rekurzív backtrackingen belül megírt iteratív backtrackinggel, vagy indirekt rekurzióval.