Problem A. Üzletek

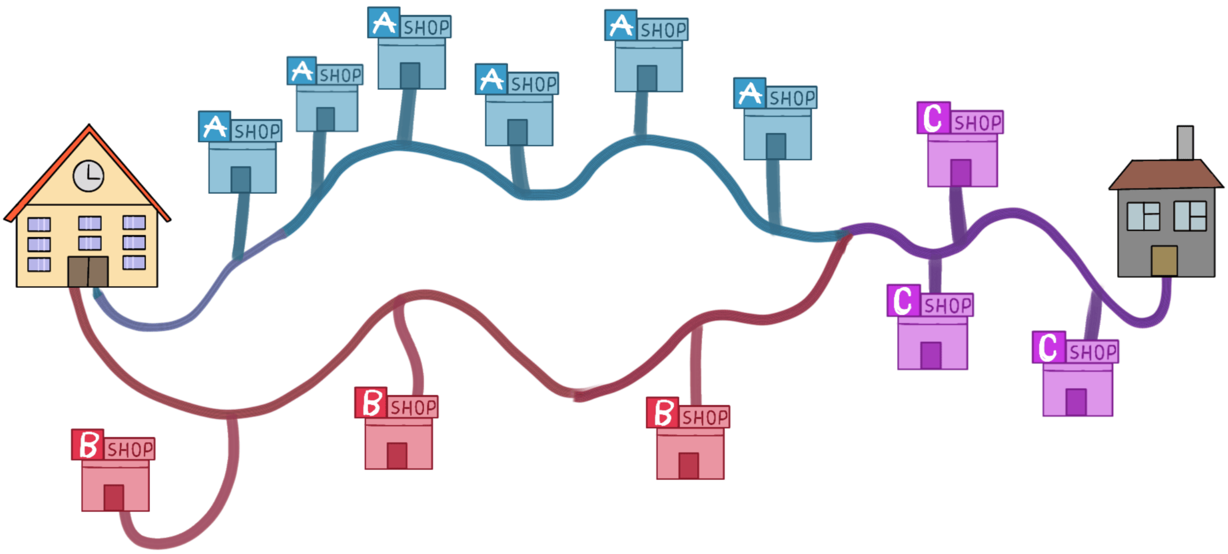
Time limit: 1 second
Memory limit: 256 megabytes

Ma péntek van. Anton végre két napig szabad az iskolából! Anton két módon mehet haza:

- az első úton haladva Anton meglátogathat a üzletet;
- a második úton haladva Anton meglátogathat b üzletet.

Anton tudja, hogy összesen c olyan üzlet van, amely mindkét úton (mindkét helyen) található. Anton kíváncsi, hogy összesen hány üzlet van a városában? Tegyük fel, hogy a városban nincsenek más üzletek ezeken kívül.

Segíts neki válaszolni erre a kérdésre.



Fent látható az egyik lehetséges útválasztás a második példában.

Input

Az első sor három egész számot tartalmaz a, b, c $(0 \le c \le a, b \le 100)$ — az üzletek száma az első úton, a második úton és mindkettőn.

Output

Egyetlen egész számot kell kiírni — a város összes üzletének számát.

Examples

standard input	standard output
1 2 0	3
9 6 3	12

Problem B. Tömb létrehozása

Time limit: 1 second
Memory limit: 256 megabytes

Sofiia Antonnak egy **számjegyeket** tartalmazó tömböt adott! Habár ez a tömb nem az első volt, amit látott, mégsem találta kevésbé érdekesnek. A tömbbel játszva észre sem vette, hogyan törte össze olyan állapotba, amelyben már nem tudta visszaállítani az eredetit.

Nagyon fel volt háborodva, mert szinte végtelen módja volt annak, hogy összeállítsa az eredeti tömböt. Azonban emlékezett az ajándék érdekes tulajdonságára: $\sum_{i=1}^{n}\sum_{j=i}^{n} \text{concat}(a_i, a_j)$, ami azt jelenti, hogy az összes elempár összefűzésének összege **maximum** az összes lehetséges tömb közül, amelyek ugyanazokból az elemekből állnak, mint az ajándék.

Más szóval, vesszük az összes i és j pozíció párt, ahol j nem balra van i-től ($i \le j$). És hozzáadjuk az összeghez $\overline{a_i a_j}$-t, ahol $\overline{ab}$ azt jelenti, hogy az a és b számokat sorrendben írjuk le (vagy $10 \cdot a + b$). Ezt nevezzük a a és b összefűzésének.

Például, ha Antonnak lenne egy tömbje $[1, 0, 3]$, akkor az összeg egyenlő lenne $\overline{a_1a_1} + \overline{a_1a_2} + \overline{a_1a_3} + \overline{a_2a_2} + \overline{a_2a_3} + \overline{a_3a_3} = 11 + 10 + 13 + 00 + 03 + 33 = 70$.

Segíts Antonnak és nyomtass egy olyan tömböt, amelynek ez a tulajdonsága van. Ha több válasz is lehetséges, bármelyiket ki lehet írni.

Input

Az első sor tartalmazza a 10 egész számot $c_0, c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9$ ($0 \le c_i \le 50$) — ahol c_i megfelel a számjegyek i számának az eredeti tömbben.

Biztosított, hogy az összes szám nagyobb, mint nulla.

Output

Nyomtass egy olyan tömböt, amely $c_0 + c_1 + c_2 + c_3 + c_4 + c_5 + c_6 + c_7 + c_8 + c_9$ elemet tartalmaz, és ugyanazokkal a tulajdonságokkal rendelkezik, mint a Sophia által adott tömb.

Examples

standard input	standard output
0 0 0 0 0 2 0 0 0 0	5 5
1 0 1 1 0 0 0 0 0 0	3 2 0

Note

A második példában ilyen lehetséges tömbök vannak:

1. $[0, 2, 3]$, az összeg egyenlő $\overline{a_1a_1} + \overline{a_1a_2} + \overline{a_1a_3} + \overline{a_2a_2} + \overline{a_2a_3} + \overline{a_3a_3} = 00 + 02 + 03 + 22 + 23 + 33 = 83$;
2. $[0, 3, 2]$, az összeg egyenlő $\overline{a_1a_1} + \overline{a_1a_2} + \overline{a_1a_3} + \overline{a_2a_2} + \overline{a_2a_3} + \overline{a_3a_3} = 00 + 03 + 02 + 33 + 32 + 22 = 92$;
3. $[2, 0, 3]$, az összeg egyenlő $\overline{a_1a_1} + \overline{a_1a_2} + \overline{a_1a_3} + \overline{a_2a_2} + \overline{a_2a_3} + \overline{a_3a_3} = 22 + 20 + 23 + 00 + 03 + 33 = 101$;
4. $[2, 3, 0]$, az összeg egyenlő $\overline{a_1a_1} + \overline{a_1a_2} + \overline{a_1a_3} + \overline{a_2a_2} + \overline{a_2a_3} + \overline{a_3a_3} = 22 + 23 + 20 + 33 + 30 + 00 = 128$;
5. $[3, 0, 2]$, az összeg egyenlő $\overline{a_1a_1} + \overline{a_1a_2} + \overline{a_1a_3} + \overline{a_2a_2} + \overline{a_2a_3} + \overline{a_3a_3} = 33 + 30 + 32 + 00 + 02 + 22 = 119$;
6. $[3, 2, 0]$, az összeg egyenlő $\overline{a_1a_1} + \overline{a_1a_2} + \overline{a_1a_3} + \overline{a_2a_2} + \overline{a_2a_3} + \overline{a_3a_3} = 33 + 32 + 30 + 22 + 20 + 00 = 137$.

Problem C. Mindenki Szereti a Permutációkat

Time limit: 1 second
Memory limit: 256 megabytes

Egy n hosszúságú permutáció egy n hosszúságú tömb, amely tartalmazza az összes egészet 1-től n-ig, és minden eleme páronként különböző.

Anton, aki gyermekkorát tömbökkel játszva töltötte, most érdekesebb tömbök tanulmányozására tért át — permutációkra. Amikor értekezését írta, nagyon nehéz feladattal szembesült.

Adott egy n hosszúságú permutáció, p, és egy egész szám, k. Úgy döntött, hogy létrehoz egy kétdimenziós tömböt, a mérettel $(k+1) \times n$.

1. $a_{0j} = j$ minden j-re $(1 \leq j \leq n)$;
2. $a_{ij} = a_{(i-1)p_j}$ minden i-re $(1 \leq i \leq k)$ és j-re $(1 \leq j \leq n)$.

Legyen $p = [5, 3, 1, 4, 2]$ és $k = 3$, ekkor az alábbi tömböt kapjuk.

a_{ij}	$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$
$i = 0$	1	2	3	4	5
$i = 1$	5	3	1	4	2
$i = 2$	2	1	5	4	3
$i = 3$	3	5	2	4	1

Minden x-re $(1 \leq x \leq n)$, szeretné tudni az összes olyan j összegét, ahol $a_{ij} = x$, ahol $1 \leq i \leq k$. Más szóval, szeretné megtalálni k szám összegét — a szám x indexeit minden a_i-ben.

Tekintsük az előző példát. Ha $x = 1$, akkor a válasz $3 + 2 + 5 = 10$ lesz.

Némi tanakodás és egyszerű ötletek után Anton gyorsan megoldotta ezt a problémát. Most azt szeretné tudni, hogy te is meg tudod-e oldani.

Input

A bemenet első sorában két egész szám található, n, k $(1 \leq n \leq 10^6, 1 \leq k \leq 10^9)$ — a permutáció hossza és az ismétlési műveletek száma.

A második sor tartalmazza a permutációt, p $(1 \leq p_i \leq n)$.

Output

Nyomtass ki n egész számot, ahol az i-edik szám a $x = i$ tartozó válasz.

Scoring

1. (8 pont): $k = 1$;
2. (17 pont): $p_i = i$;
3. (26 pont): $n \leq 2000, k \leq 2000$;
4. (28 pont): $n \leq 2000$, bármely i és j esetén létezik egy k, úgy hogy $p[p[p \ldots p[i] \ldots]] = j$, ahol a beágyazottság k-szor történik;
5. (9 pont): bármely i és j esetén létezik egy k, úgy hogy $p[p[p \ldots p[i] \ldots]] = j$, ahol a beágyazottság k-szor történik;
6. (12 pont): további korlátozás nélkül.

Examples

standard input	standard output
3 2 2 1 3	3 3 6
5 3 5 3 1 4 2	10 9 8 12 6

Problem D. Vakond

Time limit: 1 second
Memory limit: 256 megabytes

Egy nehéz kutatás után Anton úgy döntött, hogy kikapcsolódik a vidéki házában. Gyönyörű kertje van sok különböző virággal. De, ó ne, megérkezésekor jelentős számú lyukat látott a földön. Ez egy vakond!

Most, egy ásóval felfegyverkezve, Anton várni fogja a vakondot. A vakond bármelyik lyukból előbukkanhat. Anton olyan pozíciót szeretne választani, hogy a legrosszabb esetben a lehető **minimális** idő alatt fussa le a vakondot.

A kertet egy $n \times m$ mátrixként lehet reprezentálni, ahol n a sorok száma, és m az oszlopok száma. A sorokat felülről lefelé, 1-től n-ig számozzuk. Az oszlopokat balról jobbra, 1-től m-ig számozzuk. Így a $(1; 1)$ indexű cella a bal felső sarokban található.

A kert minden cellája, $a_{i,j}$, leírja a cella állapotát:

- $a_{i,j} =$ "`.`" — ez a cella nem tartalmaz virágokat vagy lyukakat;
- $a_{i,j} =$ "`F`" — ez a cella virágokat tartalmaz;
- $a_{i,j} =$ "`H`" — ez a cella lyukat tartalmaz.

Anton tudja, hogy a lyukak száma nem haladja meg a 100-at.

Mint olyan, aki sok időt fektetett ezekbe a virágokba, a szíve nem bírja elviselni, hogy letapossa azokat. Ezért olyan utat kell találnia, amely nem halad át rajtuk.

Bármikor Anton át tud menni a (x, y) pozícióból az alábbi pozíciók egyikére: $(x-1, y)$, $(x+1, y)$, $(x, y-1)$, $(x, y+1)$, feltéve, hogy az új pozíció nem tartalmaz virágokat, és a kertben van.

Keressük meg az összes olyan pozíciót $(x; y)$, ahonnan Anton a legrosszabb esetben a lehető legkevesebb idő alatt fog futni a vakondhoz.

Input

Az első sor két egész számot tartalmaz, n, m $(1 \le n \cdot m \le 2 \cdot 10^5)$ — a kert hosszát és szélességét.

A következő n sor mindegyike m karaktert tartalmaz — a kert leírása.

Biztosított, hogy minden olyan cellából, amely nem tartalmaz virágokat, elérhető másik olyan cella, amely nem tartalmaz virágokat, a cellákon keresztüli mozgással, amelyek nem tartalmaznak virágokat.

Biztosított, hogy legalább egy lyuk van, és a lyukak száma a kertben nem haladja meg a 100-at.

Output

Az első sorban egyetlen egész számot kell kiírni, x $(1 \le x \le n \cdot m)$ — az optimális pozíciók száma.

Minden következő x sorban ki kell írni az optimális pozíciókat $(x; y)$ a vakondra várásra $(1 \le x \le n$, $1 \le y \le m)$.

A pozíciókat bármilyen sorrendben ki lehet írni.

Scoring

Legyen k a lyukak száma a kertben.

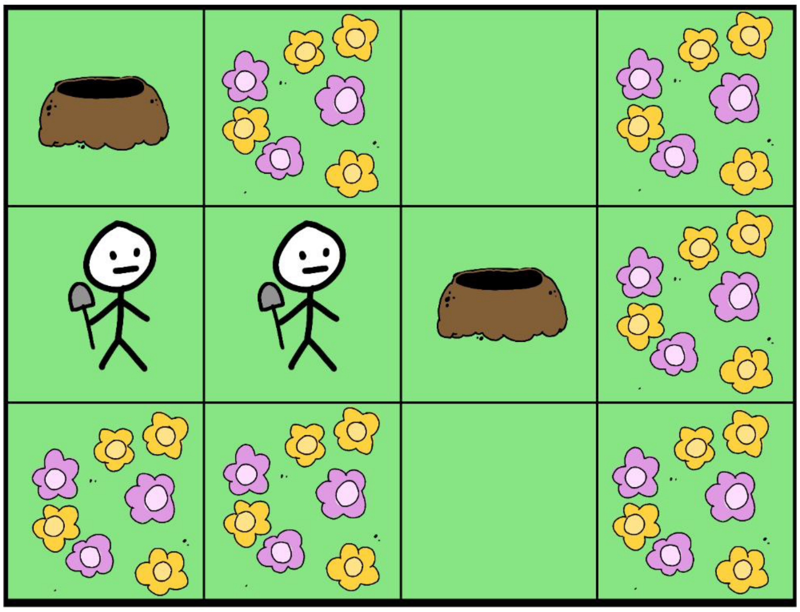
1. (6 pont): $n = 1, m = 2$;
2. (9 pont): $n = 1$;
3. (15 pont): $k = 1, n \cdot m \le 5 \cdot 10^3$;

4. (22 pont): $n \cdot m \leq 5 \cdot 10^3$;
5. (17 pont): $k = 1$;
6. (31 pont): további korlátozások nélkül.

Examples

standard input	standard output
3 4 HF.F ..HF FF.F	2 2 1 2 2
4 9 ......FFH .F..FHFF. HF....... .FHF..FFF	2 1 6 3 4

Note



Fent látható az első példa és az optimális várakozási pozíciók vannak megjelölve.

Problem E. LNKO, Összeg, Szorzat. Mi?...

Time limit: 3 seconds
Memory limit: 256 megabytes

Az író kimerítette minden kreatív készségét az előző problémákon, így Anton ebben a leírásban nem lesz kínozva. Csak egy érdekes problémát fog adni neked.

Adott egy n elemű a tömb. Továbbá adott q lekérdezés $[l; r]$. Minden lekérdezéshez keressük meg a $\text{sum}[tl; tr] \times \text{gcd}[tl; tr]$ maximális értékét az összes $(tl; tr)$ pár esetében, ahol

- $l \le tl \le tr \le r$;
- $\text{sum}[tl; tr]$ — az összes szám összege a $[tl; tr]$ szegmensben;
- $\text{gcd}[tl; tr]$ — az összes szám legnagyobb közös osztója a $[tl; tr]$ szegmensben.

Két szám legnagyobb közös osztója a b a legnagyobb pozitív egész szám x, ami osztható mind a-val, mind b-vel.

Egy halmaz legnagyobb közös osztója a legnagyobb pozitív egész szám x, ami osztható a halmaz minden elemével.

Input

Az első sor két egész számot tartalmaz, n, q $(1 \le n, q \le 2 \cdot 10^5)$ — a tömb elemeinek száma és a lekérdezések száma.

A második sor n egész számot tartalmaz, a_i $(1 \le a_i \le 6 \cdot 10^6)$ — a tömb leírása.

Minden következő q sor két egész számot tartalmaz, l, r $(1 \le l \le r \le n)$ — a lekérdezések leírása.

Output

Nyomtass ki q egész számot — a lekérdezések válaszait.

Scoring

1. (4 pont): $n \le 3$;
2. (8 pont): $n, q \le 10^3$;
3. (5 pont): $n \le 10^3$;
4. (17 pont): $n, q \le 10^5$;
5. (14 pont): $n \le 10^5$;
6. (5 pont): $a_i \le 20$;
7. (7 pont): $a_i \le 10^3$;
8. (16 pont): $l = 1$;
9. (24 pont): nincsenek további korlátozások.

Examples

standard input	standard output
3 2 3 3 2 1 3 2 3	18 9
8 6 2 4 8 8 8 2 4 16 1 8 2 5 3 4 2 4 7 7 3 6	256 192 128 128 16 192

Note

Az első példában a következő szegmensek vannak:

- $[1;1]$ — $\text{sum}[1;1] \times \text{gcd}[1;1] = 3 \times 3 = 9$;
- $[1;2]$ — $\text{sum}[1;2] \times \text{gcd}[1;2] = 6 \times 3 = 18$;
- $[1;3]$ — $\text{sum}[1;3] \times \text{gcd}[1;3] = 8 \times 1 = 8$;
- $[2;2]$ — $\text{sum}[2;2] \times \text{gcd}[2;2] = 3 \times 3 = 9$;
- $[2;3]$ — $\text{sum}[2;3] \times \text{gcd}[2;3] = 5 \times 1 = 5$;
- $[3;3]$ — $\text{sum}[3;3] \times \text{gcd}[3;3] = 2 \times 2 = 4$.