

## Problema Xidartros

Fișier de intrare      `xidartros.in`  
Fișier de ieșire      `xidartros.out`

Algoritmul Radix Sort folosit pentru sortarea unui vector format din elemente numere naturale presupune parcurgerea următorilor pași:

- se sortează elementele vectorului după ultima cifră, iar în caz de egalitate după poziția inițială din șir;
- se sortează elementele vectorului după penultima cifră, iar în caz de egalitate după poziția de la pasul precedent;
- se repetă această operație până la cifra cea mai semnificativă (de ordin maxim) din scrierea numerelor din șir.

De exemplu, pentru șirul 525, 417, 381, 291, 455, algoritmul presupune efectuarea următorilor pași:

- sortăm după ultima cifră și obținem: 381, 291, 525, 455, 417;
- sortăm după penultima cifră și obținem: 417, 525, 455, 381, 291;
- sortăm după prima cifră și obținem: 291, 381, 417, 455, 525.

Dacă în șirul inițial elementele au asociați indicii 1, 2, 3, 4, 5, iar după fiecare pas al algoritmului de sortare acești indici se scriu în ordinea corespunzătoare elementelor asociate în șirul inițial, atunci ordinea indicilor după fiecare etapă va fi:

- 3, 4, 1, 5, 2;
- 2, 1, 5, 3, 4;
- 4, 3, 2, 5, 1.

Observăm că valorile acestor șiruri reprezintă indicii inițiali ai vectorului pe care dorim să îl sortăm. Aceștia formează astfel, la fiecare pas, o permutare a numerelor de la 1 la  $N$ .

Asemănător, șirul poate să fie sortat în orice bază de numerație  $B$  dacă transformăm numerele în baza  $B$  și apoi aplicăm același procedeu. De exemplu, dacă șirul pe care dorim să îl sortăm este 6, 8, 5 și dorim să apelăm Radix Sort în baza 2, numerele vor fi transformate în 110(6), 1000(8) și 101(5). În continuare vom obține următoarele șiruri pentru fiecare bit:

- sortăm după ultimul bit: 110, 1000, 101;
- sortăm după penultimul bit: 1000, 101, 110;
- sortăm după antepenultimul bit: 1000, 101, 110;
- sortăm după cel mai semnificativ bit: 101, 110, 1000.

Prin urmare, permutările obținute sunt, în ordine:

- 1, 2, 3;
- 2, 3, 1;
- 2, 3, 1;
- 3, 1, 2.

## Cerință

Pisica Pia are un șir inițial de lungime  $N$  asupra căruia a apelat algoritmul Radix Sort în baza  $B$ . Din păcate, sora ei Mitzu s-a jucat cu șirul Piei și acesta a fost pierdut, dar Pia în continuare are permutările obținute la fiecare pas în urma apelării algoritmului. Ajutați-o pe Pia să descopere un posibil șir inițial.

## Date Intrare

Pe prima linie a fișierului de intrare se află numărul  $T$ , reprezentând numărul de teste din problemă. Prima linie a fiecărui test va conține 3 numere naturale:  $N$  - reprezentând numărul elementelor din șir,  $B$  - reprezentând baza folosită în apelarea algoritmului de Radix Sort și  $K$  - reprezentând numărul de pași ai algoritmului. Următoarele  $K$  linii conțin câte  $N$  numere naturale reprezentând permutarea indicilor inițiali ai elementelor din șir după fiecare pas al algoritmului.

## Date Ieșire

În fișierul de ieșire se vor afișa  $T$  linii, linia  $i$  conținând răspunsul pentru cel de al  $i$ -lea test. Dacă testul admite soluție, acesta va conține  $N$  numere naturale reprezentând elementele șirului inițial (scrise în baza 10), astfel încât, aplicând algoritmul Radix Sort în baza dată în test acestui șir, permutările indicilor elementelor din șir după fiecare pas al algoritmului să fie aceleași cu cele date în fișierul de intrare. Dacă testul în schimb nu admite soluție, pe linia  $i$  se va afișa  $-1$ .

## Restricții

- $1 \leq N, T \leq 10^6$
- $2 \leq B \leq 10^9$
- $1 \leq K \leq 64$
- Numărul total de elemente citite în input din toate permutările nu depășește  $10^6$ . (Suma de  $N * K$  din fiecare test  $\leq 10^6$ )
- Pentru orice test care admite soluție, există o soluție care conține elemente cuprinse între 0 și  $10^{18}$  inclusiv. **Se admite orice soluție corectă care conține elemente în acest interval.**

#	Punctaj	Restricții
1	6	$N = 4, 1 \leq T, MAXVAL \leq 30$ , pentru fiecare dintre cele $T$ teste. $MAXVAL$ în acest caz sugerează că pentru orice test care admite soluție, există o soluție care conține numai valori până în 30.
2	6	$N \leq B$ , pentru fiecare dintre cele $T$ teste
3	11	$K = 1$ , pentru fiecare dintre cele $T$ teste
4	13	$K = 2$ , pentru fiecare dintre cele $T$ teste
5	17	$B = 2$ , pentru fiecare dintre cele $T$ teste
6	19	$B = 10$ , pentru fiecare dintre cele $T$ teste
7	28	Restricții inițiale.

## Exemple

xidartros.in	xidartros.out
3	525 417 381 291 455
5 10 3	6 8 5
3 4 1 5 2	-1
2 1 5 3 4	
4 3 2 5 1	
3 2 4	
1 2 3	
2 3 1	
2 3 1	
3 1 2	
3 2 1	
3 2 1	

## Explicație

Pentru primele 2 teste, vezi explicațiile din enunțul problemei. Pentru al 3-lea test, nu există niciun șir care, sortat cu Radix Sort în baza 2, să fie sortat într-o singură instanță și să obținem permutarea 3, 2, 1.