

## Problema Teze

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

Profesorul de informatică trebuie să corecteze tezele a  $m$  elevi. Elevii au avut de rezolvat  $n$  probleme în teză, numerotate de la 1 la  $n$ . Fiecare elev a rezolvat toate problemele, deci profesorul are de corectat în total  $m \times n$  probleme. El observă că luând lucrările la rând și corectând toate problemele din prima lucrare, apoi toate problemele din a doua lucrare, ș.a.m.d., nu obține întotdeauna cel mai mic timp total de corectare a tezelor. Profesorul are următoarea metodă de corectură:

- Toate tezele se vor corecta în  $f$  faze ( $1 \leq f \leq n$ ).
- În fiecare fază  $i$  se va decide asupra unei submulțimi de probleme care nu au fost deja corectate,  $S_i$ . Aceste probleme se vor corecta în toate tezele înainte să se revină la prima teză.
- Formal, alegem  $S_i \subseteq \{1, \dots, n\}$  astfel încât:
  - $S_i \cap S_j = \emptyset, \forall i, j \in \{1, \dots, f\}, i \neq j$
  - $S_1 \cup \dots \cup S_f = \{1, \dots, n\}$

La începerea corectării fiecărei teze, trebuie identificat numele elevului, proces care durează exact  $p$  secunde de fiecare dată, chiar dacă se revine la aceeași teză de mai multe ori.

După începerea corectării unei teze, căutarea fiecărei probleme durează  $k$  secunde. Corectarea primei probleme din submulțimea aleasă durează  $t_1$  secunde, corectarea celei de-a doua probleme durează  $t_2$  secunde ș.a.m.d. Se garantează că  $t_1 < t_2 < \dots < t_n$ . De fiecare dată când se revine la o anumită teză și se reîncepe corectarea ei cu o altă submulțime de probleme, corectarea primei probleme din submulțime va dura din nou  $t_1$  secunde.

Valoarea  $t_1$  va fi dată în fișierul de intrare împreună cu valorile  $d_0, \dots, d_{q-1}$ , din care se vor obține celelalte valori din șirul  $t$  prin formula:  $t_i = t_{i-1} + d_{i \bmod q}, \forall i \in \{2, \dots, n\}$ , unde  $x \bmod y$  semnifică restul împărțirii lui  $x$  la  $y$ .

### Cerință

Să se determine timpul minim în care pot fi corectate cele  $m$  lucrări.

### Date de intrare

Pe prima linie a fișierului de intrare `teze.in` se află numerele  $n, m, p$  și  $k$ , despărțite prin câte un spațiu. Pe a doua linie se află două valori  $t_1$  și  $q$ , despărțite printr-un spațiu. Pe următoarele  $q$  linii se află valorile  $d_0, \dots, d_{q-1}$ .

### Date de ieșire

În fișierul de ieșire `teze.out` se va scrie un singur număr, reprezentând numărul de secunde în care se pot corecta tezele mod  $10^9 + 7$ .

### Restricții

- $1 \leq n \leq 1\,500\,000\,000$ , unde  $n$  este numărul problemelor din teze
- $1 \leq m, q, k \leq 1\,000$ , unde  $m$  este numărul elevilor,  $q$  este lungimea șirului  $d$ , iar  $k$  reprezintă numărul de secunde necesar căutării unei probleme
- $1 \leq p \leq 10^{10}$ , unde  $p$  este numărul de secunde necesar identificării numelui unui elev
- $1 \leq t_1, d_i \leq 10$

#	Punctaj	Restricții
1	5	$n \leq 25$
2	10	$n \leq 50$
3	20	$n \leq 10\,000$
4	30	$n \leq 10^7$
5	7	$q = 1$
6	8	$q = 2$
7	20	fără alte restricții suplimentare

## Exemple

teze.in	teze.out	Explicații
2 10 5 2 1 1 2	130	<p>Avem două probleme în teză, iar <math>t_1 = 1</math>, <math>t_2 = t_1 + d_0 = 1 + 2 = 3</math>. Dacă le corectăm împreună, timpul de corectare pentru fiecare teză va fi: 5 secunde pentru găsirea numelui elevului, 2 secunde pentru găsirea primei probleme, 1 secundă pentru corectarea primei probleme, 2 secunde pentru găsirea problemei a doua și 3 secunde pentru corectarea problemei a doua, deci în total <math>5 + 2 + 1 + 2 + 3 = 13</math> secunde pentru fiecare dintre cele 10 teze, adică 130 de secunde în total.</p> <p>O altă posibilitate este să corectăm prima dată problema 1 în toate tezele, iar pe urmă să corectăm problema 2 în toate tezele. Pentru corectarea primei probleme avem: 5 secunde începerea corectării, 2 secunde găsirea problemei și 1 secundă corectarea problemei, în total <math>5 + 2 + 1 = 8</math> pentru o teză, în total 80 de secunde pentru toate tezele. Pentru corectarea problemei 2 avem exact același calcul, rezultă în total 160 de secunde, deci prima metodă a fost mai eficientă și posibilitate mai bună nu există.</p>