

## Gourmet<sup>\*</sup> 1: Râme pane



Chef Constantin a primit comandă de  $M$  râme pane, specialitatea casei. El se uită în cufărul de râme și observă că are la dispoziție  $N$  râme, de lungimi întregi  $r_1, r_2, \dots, r_N$ .

Chef iscusit, Constantin poate tăia râmele în bucăți mai mici de lungime **întreagă**. Apoi acesta prăjește câte o rămă (sau bucată de rămă) pentru fiecare dintre cei  $M$  oaspeți. Satisfacția unui oaspete este dată de lungimea râmei primite iar satisfacția lui Constantin, om empatic, este minimul satisfacțiilor tuturor clienților săi.

Dacă taie și distribuie râmele în mod optim, care este satisfacția maximă pe care o poate avea Chef Constantin?

### Date de Intrare

Pe prima linie se dau două numere  $N$  și  $M$ , cu semnificația din enunț.

Următoarea linie conține  $N$  numere:  $r_1, r_2, \dots, r_N$ .

### Date de Ieșire

Pe singura linie se va afișa un singur număr, satisfacția maximă a lui Chef Constantin. Dacă nu există nicio împărțire posibilă, atunci satisfacția lui este 0.

### Constrângeri

- $1 \leq N, M \leq 10^5$ .
- $\forall 1 \leq i \leq N, 1 \leq r_i \leq 10^6$ .

### Subtask-uri

1. (15 puncte)  $M = 2$ .

---

<sup>\*</sup>Preparatele gourmet se definesc ca fiind acele preparate de bună calitate, pregătite cu foarte multă grijă și prezentate într-un mod artistic.

2. (15 puncte)  $N = 2$ .
3. (40 puncte)  $1 \leq N, M \leq 10^3, \quad \forall 1 \leq i \leq N, 1 \leq r_i \leq 10^3$ .
4. (30 de puncte) Nicio constrângere suplimentară.

### Exemplu

| Input Standard ( <i>cin</i> )        | Output Standard ( <i>cout</i> ) |
|--------------------------------------|---------------------------------|
| 1 2<br>255                           | 127                             |
| 10 6<br>15 43 72 59 18 8 24 97 61 27 | 43                              |

## Gourmet 2: Prăjitură cu mujdei



Chef Alexandru a primit o comandă pentru specialitatea casei: o porție zdravănă de prăjitură cu mujdei.

Pentru prăjitură, el are nevoie de exact  $X$  litri de mujdei. Chef Alexandru are la dispoziție două castroane  $A$  și  $B$ , de capacitate  $C_A$ , respectiv  $C_B$  litri, ambele inițial goale.

Chef Alexandru poate face următoarele operații:

- **UMPLE**  $[A/B]$  - umple complet castronul  $A$  sau  $B$  cu  $C_A$ , respectiv  $C_B$  litri de mujdei;
- **GOLESTE**  $[A/B]$  - golește complet castronul  $A$  sau  $B$ ;
- **VARSA**  $[A/B]$  **IN**  $[B/A]$  - varsă cât mai mult mujdei posibil din castronul  $A$  (sau  $B$ ) în castronul  $B$  (sau  $A$ ). Dacă castronul sursă are  $x$  litri de mujdei și castronul destinație are  $y$  litri liberi, atunci Chef Alexandru va varsa exact  $\min(x, y)$  litri.

Chef Alexandru vă cere ajutorul: găsiți o serie de cel mult  $10^6$  operații în urma cărora se vor afla exact  $X$  litri de mujdei într-unul dintre cele două castroane.

**Atenție:** Orice soluție validă este acceptată, nu este necesar să minimizați lungimea acesteia!

### Date de intrare

Pe singura linie din fișierul de intrare se dau trei numere naturale  $C_A$ ,  $C_B$  și  $X$  separate prin câte un spațiu, având semnificația din enunț.

### Date de ieșire

Dacă nu există soluție, atunci pe singura linie din fișierul de ieșire afișați "IMPOSIBIL".

Dacă există o soluție, atunci afișați pe prima linie numărul de operații necesare, iar pe următoarele linii, în ordine, afișați operațiile efectuate.

### Constrângeri

- $1 \leq X < C_A, C_B \leq 100.000$ .

## Subtask-uri

1. (20 de puncte)  $X = 1, C_A = 2$  și  $C_B \leq 10$ .
2. (20 de puncte)  $1 \leq C_A, C_B \leq 10$ .
3. (30 de puncte)  $1 \leq C_A, C_B \leq 1000$ .
4. (30 de puncte) Nicio constrângere suplimentară.

## Exemplu

| Input Standard ( <i>cin</i> ) | Output Standard ( <i>cout</i> )                                                 |
|-------------------------------|---------------------------------------------------------------------------------|
| 3 2 1                         | 2<br>UMPLE A<br>VARSA A IN B                                                    |
| 1110 10101 110                | IMPOSIBIL                                                                       |
| 10 6 4                        | 6<br>UMPLE B<br>VARSA B IN A<br>UMPLE A<br>UMPLE B<br>GOLESTE B<br>VARSA A IN B |

## Gourmet 3: Răcituri cu pireu



Teo se duce la cantina lui favorită (cea de la ASE) ca să guste din noua gamă de alimente - răcituri cu pireu.

La bufetul de răcituri, el are la alegere  $R$  tipuri de răcituri. Al  $i$ -lea tip ( $0 \leq i < R$ ) costă  $r_i$  lei. Pentru fiecare tip de răcitură, Teo, cu o probabilitate de  $\frac{1}{2}$ , își pune în farfurie o răcitură.

La bufetul de pireu, el are la alegere  $P$  tipuri de pireu. Pireul se vinde la volum, așa că pentru al  $i$ -lea tip de pireu ( $0 \leq i < P$ ), Teo se poate servi cu pireu în valoare de  $0, 1, \dots, p_i$  lei. Pentru fiecare tip de pireu, Teo se servește uniform aleatoriu cu o cantitate permisă de pireu.

Cantina oferă următoarea reducere: dacă prețul total al alimentelor alese de Teo este strict mai mare decât suma de  $X$  lei, atunci Teo primește o reducere de 50%.

Care este, în medie, prețul plătit de Teo?

### Date de intrare

Pe prima linie din fișierul de intrare se află numerele  $R$ ,  $P$  și  $X$  cu semnificația din enunț.

Pe a doua linie se află  $R$  numere întregi:  $r_0, r_1, \dots, r_{R-1}$ .

Pe a treia linie se află  $P$  numere întregi:  $p_0, p_1, \dots, p_{P-1}$ .

### Date de ieșire

Pe singura linie din fișierul de ieșire afișați răspunsul, modulo  $10^9 + 7$ .

Dacă răspunsul este egal cu fracția  $\frac{A}{B}$ , afișați numărul  $A * B^{-1} \pmod{10^9 + 7}$ .

### Constrângeri

- $1 \leq P, R \leq 100$ .
- $1 \leq p_i, r_i \leq 10^3$ .
- $1 \leq X \leq 2 * 10^5$ .

## Subtask-uri

1. (20 de puncte)  $P = 0, R \leq 20$  și  $X = 2 \cdot 10^5$  (niciodată nu se aplică reducerea).
2. (20 de puncte)  $P = 0, R \leq 20$ .
3. (20 de puncte)  $P = 0$ .
4. (20 de puncte)  $\forall 0 \leq i < P, \quad p_i \leq 5$ .
5. (20 de puncte) Nicio constrângere suplimentară.

## Atașamente

În secțiunea ”Downloads” a interfeței de concurs puteți descărca un fișier C++ care implementează operații de bază ale aritmeticii modulare. Acesta este util mai ales dacă nu ați mai lucrat cu noțiunea de invers modular.

## Exemplu

| Input Standard ( <i>cin</i> )                                             | Output Standard ( <i>cout</i> ) |
|---------------------------------------------------------------------------|---------------------------------|
| 5 3 10<br>2 7 6 2 6<br>6 7 10                                             | 142806425                       |
| 10 10 100<br>26 27 72 11 6 30 60 90 28 15<br>60 68 38 48 22 65 19 75 17 1 | 413422235                       |

## Gourmet 4: Cornișoni afumați



În anul de glorie 2051 ChatGPT 42, maestru arheolog, a descoperit o carte de rețete, o rămășiță a civilizației umane, exterminată de aproape 26 de ani.

Maestrul vrea acum să recreeze o delicată umană: cornișoni afumați. El are de urmat  $N$  pași, dar este puțin derutat: rețeta, ca orice rețetă scrisă în 2024 care se respectă, vorbește de copilăria autorului, apoi de ingrediente, apoi de o pățanie haioasă a autorului etc. Pentru a putea recrea rețeta, roboțelul nostru are de sortat pașii și vă cere ajutorul.

Cei  $N$  pași de urmat, reprezentați prin numere de la 0 la  $N - 1$ , și salvați în vectorul  $V$  (elementele  $V[0], \dots, V[N - 1]$ ), se află într-o ordine aleatorie (o permutare) și trebuie sortați în ordine crescătoare. Roboțelul poate ține minte 5 numere în *variabilele*  $A, B, C, D$  și  $E$  (inițial, toate variabilele au valoarea 0). Adicional, variabilele constante  $Z$  și  $N$  conțin valoarea 0, respectiv  $N$ , numărul de pași de urmat.

Instrucțiunile pe care le poate efectua robotul sunt:

- **IF\_LESS\_GOTO  $r1$   $r2$   $x$**  – dacă valoarea variabilei  $r1$  este mai mică decât valoarea variabilei  $r2$ , atunci următoarea operație pe care o execută robotul este  $x$ ;
- **IF\_DIFF\_GOTO  $r1$   $r2$   $x$**  – dacă valoarea lui  $r1$  este diferită de valoarea lui  $r2$ , atunci următoarea operație pe care o execută robotul este  $x$ ;
- **IF\_SAME\_GOTO  $r1$   $r2$   $x$**  – dacă valoarea lui  $r1$  este egală cu valoarea lui  $r2$ , atunci următoarea operație pe care o execută robotul este  $x$ ;
- **ASSIGN  $r1$   $r2$**  – se salvează în  $r1$  valoarea lui  $r2$ . Variabila  $r1$  trebuie să fie **ne-constantă** (i.e. diferită de  $Z$  și  $N$ ).
- **INC  $r$**  și **DEC  $r$**  – se crește sau se scade valoarea salvată în variabila **ne-constantă**  $r$ .
- **PLOAD  $r$**  – în variabila **ne-constantă**  $r$  se salvează pasul de la poziția  $r$  ( $r$  devine  $V[r]$ ). Valoarea lui  $r$  trebuie să fie cuprinsă între 0 și  $N - 1$ .
- **PSWAP  $r1$   $r2$**  – se interschimbă valorile din  $V[r1]$  și  $V[r2]$  (i.e.  $V[r1]$  devine  $V[r2]$  și vice-versa). Valorile din  $r1$  și  $r2$  trebuie să fie cuprinse între 0 și  $N - 1$  inclusiv.

- **END** – se termină execuția programului.

După ce execută o operație, robotul execută următoarea operație, mai puțin în cazul în care operația specifică altceva. Dacă a ajuns la sfârșitul operațiilor sau s-a executat operația **END**, execuția se termină.

## Date de intrare

Pe unica linie din fișierul de intrare se află numărul  $N$ , lungimea vectorului  $V$ .

## Date de ieșire

Se afișează un program care să ordoneze vectorul  $V$  cu cei  $N$  pași în ordine crescătoare. Dacă există mai multe soluții posibile, atunci se va accepta oricare dintre ele.

**Atenție:** pentru formatul exact în care trebuie afișate operațiile vedeți exemplele.

## Constrângeri

- $2 \leq n \leq 100.000$ .
- Programul generat nu poate efectua mai mult de  $5 \cdot 10^6$  operații.
- Programul generat nu poate avea mai mult de 1.000 de instrucțiuni.

## Subtask-uri

1. (10 de puncte)  $N = 3$ .
2. (15 de puncte)  $N = 4$ .
3. (15 de puncte)  $N \leq 15$ .
4. (20 de puncte)  $N \leq 100$ .
5. (20 de puncte)  $N \leq 5.000$ .
6. (20 de puncte) Nicio constrângere suplimentară.

## Atașamente

Pentru a simplifica testarea soluțiilor, în secțiunea ”*Downloads*” a interfeței de concurs puteți descărca un validator pentru soluția voastră.

## Exemplu

| Input Standard ( <i>cin</i> ) | Output Standard ( <i>cout</i> )                                                                                                                                                                   |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2                             | 0. INC A<br>1. PLOAD A<br>2. PLOAD B<br>3. IF_LESS_GOTO B A 5<br>4. END<br>5. ASSIGN A Z<br>6. ASSIGN B Z<br>7. INC B<br>8. PSWAP A B                                                             |
| 2                             | 0. ASSIGN A N<br>1. DEC A<br>2. ASSIGN B Z<br>3. ASSIGN C A<br>4. ASSIGN D B<br>5. PLOAD C<br>6. PLOAD D<br>7. IF_DIFF_GOTO N Z 9<br>8. END<br>9. IF_LESS_GOTO C D 11<br>10. END<br>11. PSWAP A B |
| 2                             | 0. INC B<br>1. PLOAD A<br>2. PLOAD B<br>3. IF_LESS_GOTO A B 6<br>4. INC D<br>5. PSWAP C D<br>6. END                                                                                               |

Considerăm al doilea exemplu ( $N = 2$ ):

```

0. ASSIGN A N
1. DEC A
2. ASSIGN B Z
3. ASSIGN C A
4. ASSIGN D B
5. PLOAD C
6. PLOAD D
7. IF_DIFF_GOTO N Z 9
8. END
9. IF_LESS_GOTO C D 11
10. END
11. PSWAP A B

```

Presupunem că valoarea permutării este  $V = \{ 1, 0 \}$ . Operațiile efectuate de robot sunt

următoarele:

1. Linia executată este 0. ASSIGN A N.  
 $V = \{ 1, 0 \}$   
 $A = 2, B = 0, C = 0, D = 0, E = 0$
2. Linia executată este 1. DEC A.  
 $V = \{ 1, 0 \}$   
 $A = 1, B = 0, C = 0, D = 0, E = 0$
3. Linia executată este 3. ASSIGN C A.  
 $V = \{ 1, 0 \}$   
 $A = 1, B = 0, C = 1, D = 0, E = 0$
4. Linia executată este 4. ASSIGN D B.  
 $V = \{ 1, 0 \}$   
 $A = 1, B = 0, C = 1, D = 0, E = 0$
5. Linia executată este 5. PLOAD C.  
 $V = \{ 1, 0 \}$   
 $A = 1, B = 0, C = 0, D = 0, E = 0$
6. Linia executată este 6. PLOAD D.  
 $V = \{ 1, 0 \}$   
 $A = 1, B = 0, C = 0, D = 1, E = 0$
7. Linia executată este 7. IF\_DIFF\_GOTO N Z 9.  
 $N$  este diferit de  $Z$  ( $2 \neq 0$ ), așa că sărim la instrucțiunea 9.  
 $V = \{ 1, 0 \}$   
 $A = 1, B = 0, C = 0, D = 1, E = 0$
8. Linia executată este 9. IF\_LESS\_GOTO C D 11.  
 $C$  este mai mic decât  $D$ , așa că sărim la instrucțiunea 11.  $V = \{ 1, 0 \}$   
 $A = 1, B = 0, C = 0, D = 1, E = 0$
9. Linia executată este 11. PSWAP A B.  
 $V = \{ 0, 1 \}$   
 $A = 1, B = 0, C = 0, D = 1, E = 0$
10. Execuția programului se oprește, iar vectorul  $V = \{0, 1\}$  este sortat.