

Secțiunea 7-8 începători

PROBLEMA 1 COMOARA

100 puncte

După ce l-a învins pe maleficul vrăjitor Rumbal, Budi a plecat într-o nouă aventură: să găsească secreta comoară a lui Mârț. Harta l-a ghidat spre caverna în care Mârț și-a ascuns averea. După ce a trecut de multe capcane, Budi se găsește în fața ușii camerei comorii. Însă aceasta este ferecată.

Pe peretele de lângă ușă apar o serie de butoane, numerotate de la stânga la dreapta de la 1 la N, și un panou cu instrucțiuni alături de K perechi de numere. Budi citește pe panou că, pentru a ajunge la comoară, pentru fiecare pereche de forma (x y) care apare, el trebuie să apese pe toate butoanele de la al x-lea până la al y-lea buton, inclusiv.

Budi observă, de asemenea, că dacă un buton apăsător este apăsător a doua oară, el revine la poziția inițială, (cea înainte de prima apăsare), și că unele butoane de pe ușă au fost deja apăstate de altcineva, înaintea lui. Pentru că este posibil ca Budi să fie nevoit să apese unele butoane de foarte multe ori, și asta i-ar lua prea mult timp, el vă cere ajutorul.

**Cerință**

Știind configurația inițială a butoanelor și conținutul panoului, să se afișeze numărul minim de butoane, precum și indicii butoanelor care, apăstate o singură dată, vor aduce configurația inițială la configurația care îi deschide lui Budi ușa spre comoară.

**Date de intrare**

Pe primul rând al fișierului *comoara.in* se află trei numere N, M, K reprezentând numărul de butoane, numărul de butoane apăstate de cealaltă persoană înainte de venirea lui Budi și numărul de intervale de pe panou. Pe următoarea linie se află M numere, care reprezintă indicii butoanelor apăstate în ordine înainte de venirea lui Budi. Pe următoarele K linii se află câte două numere x și y, care reprezintă capetele intervalelor de pe panou.

**Date de ieșire**

În fișierul *comoara.out* se vor afișa, pe un singur rând, indicii butoanelor care trebuie apăstate, o singură dată, pentru a ajunge la configurația finală.

**Restricții și precizări:**

- $2 \leq M \leq N \leq 300000$ ,  $1 \leq K \leq 100000$
- Perechile de pe panou sunt formate din două numere naturale în intervalul  $[1, N]$
- Persoana care a fost înaintea lui Budi, poate apăsa de mai multe ori un același buton, având efectul specificat în enunț.

**Exemplu**

comoara.in	comoara.out	Explicație
4 1 2 2 1 4 2 3	1 2 4	Plecând de la o configurație cu toate butoanele neapăstate, după primul interval $[1, 4]$ , toate butoanele vor fi apăstate. După al doilea interval $[2, 3]$ , butoanele 2 și 3 vor deveni neapăstate, apăstate vor rămâne doar butoanele 1 și 4, care ar reprezenta configurația finală. Știind că butonul 2 era apăsat inițial (și nu ar trebui să fie în final) trebuie apăsat și acesta. Deci butoanele care trebuie apăstate sunt 1, 2, 4.

**Timp maxim de execuție:** 1 secundă/test

**Memorie totală disponibilă:** 12 MB din care 12 MB pentru stivă

**Dimensiunea maximă a sursei:** 5 Kb

Secțiunea 7-8 începători

**PROBLEMA 2      SECVENȚA**

**100 puncte**

Fie un șir de N valori numere naturale și K, o valoare număr natural nenul.

**Cerință**

Să se afișeze pe ecran, pe linii separate, primele K subsecvențe strict crescătoare ale șirului citit, urmate de o valoare de 2000000000.

Câte valori au fost tipărite, excluzând valorile finale de 2000000000?

În a câta cea din urmă secvență se află una dintre valorile maxime tipărite, excluzând valorile finale de 2000000000?

**Date de intrare**

Fișierul *secventa.in* cu următoarea structură:

- pe prima linie valorile N și K cu semnificația din enunț.
- pe a doua linie N valori numere naturale reprezentând șirul respectiv de valori.

**Date de ieșire**

Fișierul *secventa.out* cu următoarea structură: pe primele K linii cele K secvențe din enunț urmate de santinela 2000000000. Pe linia K+1 numărul de valori tipărite. Pe linia K+2 numărul secvenței de valoare maximă.

**Restricții și precizări**

- $10 \leq N \leq 1000$ ;  $1 \leq K \leq 500$ .
- Valorile din șir sunt cuprinse între 1 și 1000.
- Numărul de subsecvențe strict crescătoare din șir este strict mai mare decât K.
- O subsecvență de numere crescătoare reprezintă un subșir de numere de pe poziții consecutive, maximal cu proprietatea dorită.

**Exemplu**

secventa.in	secventa.out	Explicație
12 3	1 2 2000000000	Pe primele 3 linii primele 3 secvențe strict crescătoare. Pe a 4 linie numărul de elemente ale celor 3 secvențe. Pe a 5 linie valoarea 2, reprezentând ultima secvență ce conține una dintre valorile maxime tipărite, în acest caz valoarea 3.
1 2 1 2 3 1 2 1 2 3 4 5	1 2 3 2000000000	
	1 2 2000000000	
	7	
	2	

**Timp maxim de execuție:** 0.1 secunde/test

**Memorie totală disponibilă:** 2MB din care 2MB pentru stivă

**Dimensiunea maximă a sursei:** 10KB

**PROBLEMA 3      ALBINUȚE**

**100 puncte**

Într-un stup de albinuțe e mare zarvă deoarece Regina Albină este foarte furioasă pe albinuțele ei care sunt leneșe și nu își mai îndeplinesc norma de lucru. Albinuțele sunt triste că au supărat-o pe Regină așa că se hotărăsc să se mobilizeze să își termine treaba și se întrec să rezolve cât mai multe „task-uri” de pe lista acesteia. Știm că în stup sunt N albinuțe:  $a_1, a_2, a_3, \dots, a_n$ , că prima albinuță

## Secțiunea 7-8 începători

rezolvă doar un task și că fiecare albinuță rezolvă de trei ori mai multe și încă patru task-uri decât albinuța dinaintea ei.

### Cerință

Regina va roagă sa răspundeți la Q întrebări de forma: "Câte sarcini a completat a P-a albinuță?". **Atenție!!** Albinuțele sunt foarte harnice, iar numărul task-urilor efectuate poate fi foarte mare. În consecință rezultatele se vor calcula modulo 100007.

### Date de intrare

Fișierul **albinute.in** cu următoarea structură: pe prima linie numărul N de albinuțe și numărul Q de întrebări, pe următoarele Q linii câte un număr P reprezentând numărul de ordine al albinuței pentru care Regina vrea sa știe numărul de task-uri efectuate.

### Date de ieșire

Fișierul **albinute.out** cu următoarea structură: Q linii pe care se vor afla răspunsurile la cele Q întrebări.

### Restricții și precizări

- $1 \leq P < N \leq 999.999.999$
- $1 \leq Q \leq 10.000$
- P-urile din întrebări se pot repeta

### Exemplu

albinute.in	albinute.out	Explicație
10 4	1	Prima albinuță completează un task, a doua $3*1+4=7$ task-uri și următoarele 25, 79, 241 și 727. Deci a 5-a și a 6-a albinuță completează 241, respectiv 727 sarcini.
1	7	
2	241	
5	727	
6		

**Timp maxim de execuție:** 0.05 secunde/test

**Memorie totală disponibilă:** 64 MB din care 64 MB pentru stivă

**Dimensiunea maximă a sursei:** 5 Kb

## PROBLEMA 4 CARGO

100 puncte

Cargourile sunt cele mai răspândite tipuri de nave maritime de transport. Ele sunt construite pentru a transporta mărfuri generale sau pot fi specializate pentru anumite tipuri de marfă. Pentru a ușura operațiile de încărcare-descărcare, mărfurile generale sunt ambalate în pachete mari, se folosesc palete de dimensiuni standard pe care se stivuiește încărcătura sau se utilizează cutii metalice standardizate de dimensiuni mari numite containere, care pot transporta greutatea de câteva zeci de tone.

Pe cargoul CONTA, cala sau magazia este organizată în 3 zone de dimensiuni diferite, în fiecare zonă putând fi depozitate un anumit număr de containere. Pe fiecare container se găsește înscrisă greutatea sa în kilograme.

## Secțiunea 7-8 începători

### Cerință

Cunoscând numărul de containere și greutatea containerelor din fiecare zonă, să se determine **combinația-echilibrată**, definită astfel: 3 containere, câte unul din fiecare zonă, pentru care diferența dintre greutatea celui mai greu și celui mai ușor container este minimă.

Se vor afișa greutatea containerelor din combinația-echilibrată de greutate maximă și numărul de combinații-echilibrate care se pot determina.

### Date de intrare

Fișierul **cargo.in** are următoarea structură:

- pe prima linie trei numere naturale **n**, **m** și **p** reprezentând numărul de containere din fiecare zonă;
- pe linia a doua **n** numere naturale nenule distincte, separate prin spațiu, reprezentând greutatea containerelor din prima zonă;
- pe linia a treia **m** numere naturale nenule distincte, separate prin spațiu, reprezentând greutatea containerelor din zona a doua;
- pe linia a patra **p** numere naturale nenule distincte, separate prin spațiu, reprezentând greutatea containerelor din zona a treia.

### Date de ieșire

Fișierul **cargo.out** are următoarea structură:

- pe prima linie greutatea containerelor din combinația-echilibrată determinată conform cerinței, scrise în ordine descrescătoare, separate printr-un spațiu;
- pe a doua linie numărul de combinații-echilibrate.

### Restricții și precizări:

- $1 \leq n, m, p \leq 1000$ ;
- greutatea unui container este de cel mult 40000 kg.

### Exemplu

cargo.in	cargo.out	Explicație
4 3 6 11 19 5 10 18 21 15 20 12 16 13 14 17	21 20 19 3	Sunt 3 combinații-echilibrate cu proprietatea din enunț (19, 18, 20), (19, 18, 17) și (19, 21, 20).

**Timp maxim de execuție:** 0.1 secunde/test

**Memorie totală disponibilă** 2 MB, din care 2 MB pentru stivă

**Dimensiunea maximă a sursei:** 5 KB