

ab**100 puncte**

Una din cele mai noi pasiuni ale lui Zăhărel este să studieze diverse proprietăți ale permutărilor. De exemplu, este interesat de permutările în care cel mai lung subșir crescător și cel mai lung subșir descrescător au lungimi date.

Cerință

Să se scrie un program care determină o permutare de lungime N în care cel mai lung subșir crescător are lungime A și cel mai lung subșir descrescător are lungime B .

Date de intrare

Fișierul de intrare `ab.in` va conține pe prima linie numerele N A B .

Date de ieșire

Fișierul de ieșire `ab.out` va conține pe prima linie N numere separate prin câte un spațiu, reprezentând o permutare care respectă condițiile de mai sus. Dacă există mai multe soluții, se va afișa cea minimă din punct de vedere lexicografic.

Restricții și precizări

- $1 \leq N, A, B \leq 30.000$
- Se garantează că mereu există soluție pentru datele de intrare
- Se numește subșir al șirului $X = (x_1, x_2, \dots, x_N)$, un șir $Y = (x_{i_1}, x_{i_2}, \dots, x_{i_M})$ cu proprietatea $1 \leq i_1 < i_2 < \dots < i_M \leq N$
- Un șir $(x_1, x_2, \dots, x_K)$ este mai mic din punct de vedere lexicografic decât un alt șir $(y_1, y_2, \dots, y_K)$ dacă există o poziție $p \leq K$, astfel încât $x_p < y_p$ și $x_1 = y_1, x_2 = y_2, \dots, x_{p-1} = y_{p-1}$
- Pentru un test se va acorda 70% din punctaj dacă permutarea afișată are cel mai lung subșir crescător de lungime A și cel mai lung subșir descrescător de lungime B , dar nu este minimă lexicografic.

Exemplu

ab.in	ab.out	Explicație
4 2 3	1 4 3 2	Cel mai lung subșir crescător are lungime 2 (1 4, 1 3 sau 1 2), iar cel mai lung subșir descrescător are lungime 3 (4 3 2). O altă soluție posibilă este 2 4 3 1, dar aceasta nu este minimă din punct de vedere lexicografic.

Timp maxim de execuție/test: 0.1 secunde