

## 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 1. PLOAD A 2. PLOAD B 3. IF_LESS_GOTO B A 5 4. END 5. ASSIGN A Z 6. ASSIGN B Z 7. INC B 8. PSWAP A B
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
2	0. INC B 1. PLOAD A 2. PLOAD B 3. IF_LESS_GOTO A B 6 4. INC D 5. PSWAP C D 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.