

Coach

100 puncte

Fișiere sursă: `coach.pas` sau `coach.c` sau `coach.cpp`

Sunteți antrenorul ciclistului Adirem Onamihs. În curând va avea loc un eveniment sportiv, iar pentru organizarea acestuia s-au amenajat **N** intersecții și **M** drumuri **bidirecționale** între aceste intersecții. Pentru fiecare drum se cunoaște numărul de minute necesare pentru parcurgerea lui. La fiecare intersecție ciclistul care trece pe acolo este obligat să servească o băutură energizantă și răcoritoare. Băutura diferă de la intersecție la intersecție și se cunoaște deja numărul de calorii ale fiecărei băuturi.

Ca mare antrenor, urmează să întocmiți un plan special pentru a-l antrena pe Adirem. Doriți ca durata traseului pe care îl alege Adirem să aibă **exact T** minute, însă nu vreți să-i plănuți întregul traseu (Adirem trebuie să își antreneze și mintea, nu numai corpul). Îi veți preciza lui Adirem intersecția de unde își începe traseul și intersecția unde îl termină. Adirem învață repede - el știe întotdeauna să aleagă traseul optim (drumul cel mai scurt dintre cele două intersecții). Pentru a-l face să meargă exact **T** minute îi veți interzice trecerea prin anumite intersecții, sub pretextul că valoarea calorică a băuturii servite în intersecția respectivă nu este benefică pentru antrenamentul lui. Astfel, îi veți preciza o limită inferioară și una superioară pentru numărul de calorii ale băuturilor pe care el are voie să le bea. Adirem nu va trece decât prin intersecțiile unde se servește o băutură cu valoare calorică între limitele date.

Cerință

Scrieți un program care să calculeze cele patru variabile din antrenamentul lui Adirem: intersecția de start, intersecția de finish, valoarea calorică minimă pe care poate să o consume și valoarea calorică maximă, astfel încât drumul cel mai scurt dintre cele două intersecții (care să respecte restricțiile) să dureze exact **T** minute.

Date de intrare

Prima linie a fișierului `coach.in` conține trei numere întregi **N**, **M** și **T** - numărul de intersecții, numărul de drumuri, respectiv timpul dorit. Următoarele **N** linii conțin câte un număr - valorile calorice (întregi între **1** și **10000** inclusiv) ale băuturilor din intersecții, în ordine (de la **1** la **N**). Următoarele **M** linii conțin câte un triplet de numere: două intersecții (numere distincte între **1** și **N**) și durata drumului dintre ele (întreg între **1** și **10000** inclusiv).

Date de ieșire

Fișierul `coach.out` va conține o linie pe care se vor afla cele patru valori găsite: nodul de start, nodul de finish, valoarea calorică minimă și valoarea calorică maximă. Nodurile vor fi întregi între **1** și **N**, iar valorile calorice vor fi întregi între **1** și **10000** (inclusiv).

Restricții și precizări

- $1 \leq N \leq 100$,
 $1 \leq M \leq 4950$,
 $1 \leq T \leq 1000000$
- Intersecțiile găsite (de start și de finish) trebuie să respecte și ele restricțiile calorice
- O băutură cu valoare calorică **x** poate fi băută dacă și numai dacă $c_{min} \leq x \leq c_{max}$, unde **c_{min}** și **c_{max}** sunt valorile calorice minime și maxime stabilite de antrenor
- Între două intersecții există maximum un drum
- Valorile calorice sunt distincte
- Există întotdeauna soluție; dacă există mai multe soluții se cere oricare dintre ele

Exemplu

<code>coach.in</code>	<code>coach.out</code>
6 9 11	3 6 20 55
40	
10	
20	
30	
60	
50	
1 2 2	
1 3 2	
1 4 4	
1 6 10	
2 3 3	
2 4 1	
4 5 1	
4 6 5	
5 6 2	

Timp maxim de executare/test:

0.5 secunde pentru Linux și 0.9 secunde pentru Windows

Patrate

100 puncte

Fișiere sursă: `patrate.pas` sau `patrate.c` sau `patrate.cpp`

Ovi este un băiețel foarte isteț căruia îi place să scrie pe asfalt cu creta și să țopăie. El desenează cu cretă roșie un dreptunghi de lățime exact 2 metri și lungime N metri, pe care îl împarte în pătrate egale de latură 1 metru, unele laturi interioare fiind desenate cu cretă roșie, iar restul laturilor interioare cu cretă albă. Ovi pornește din pătratul aflat în colțul stânga sus al dreptunghiului, sărind dintr-un pătrat în altul vecin pe linie sau coloană, cu condiția ca latura care desparte cele două pătrate să nu fie colorată în roșu. El își dorește ca prin sărituri succesive să ajungă în toate pătratele dreptunghiului, dar a observat că numai pentru anumite variante de colorare a laturilor pătratelor reușește acest lucru.

În exemplele de mai jos (cu $N=2$) liniile interioare îngroșate sunt colorate cu roșu, iar cele punctate sunt colorate cu alb. La exemplul din fig. 1, pornind din colțul stânga sus se poate ajunge în oricare alt pătrat, dar în exemplul din fig. 2 nu se poate ajunge la pătratele din partea dreaptă.

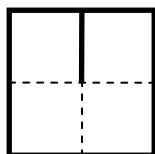


fig. 1

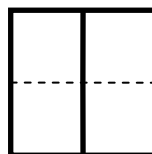


fig. 2

Cerință

Ajutați-l pe Ovi să numere câte posibilități de colorare în roșu a unor laturi interioare ale pătratelor sunt astfel încât plecând din colțul stânga sus să poată ajunge prin sărituri în oricare alt pătrat.

Date de intrare

În fișierul `patrate.in` se află un singur număr natural N ce reprezintă lungimea în metri a dreptunghiului.

Date de ieșire

În fișierul `patrate.out` veți scrie un singur număr natural (urmat de caracterul de sfârșit de linie) ce reprezintă numărul de posibilități cerut.

Restricții și precizări

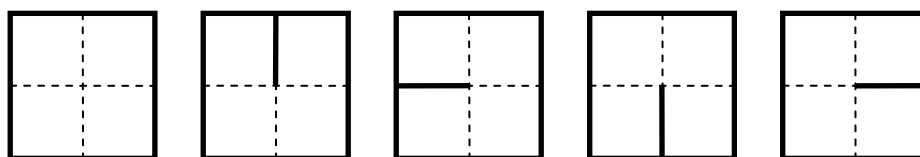
- $2 \leq N \leq 1000$

Exemplu

<code>patrate.in</code>	<code>patrate.out</code>
2	5

Explicație:

Cele 5 posibilități sunt:



Timp maxim de executare/test: 0.2 secunde pentru Linux și 0.4 secunde pentru Windows

Base3

100 puncte

Fișiere sursă: **base3.pas** sau **base3.c** sau **base3.cpp**

Se dau 3 numere scrise în baza 3 (folosind cifrele 0, 1 și 2). Se dorește găsirea unui număr N în baza 3, care să aibă un număr impar de cifre, iar cifra de pe poziția din mijloc să aibă valoarea 1. Acest număr N trebuie obținut prin concatenarea celor trei numere date; în această concatenare, fiecare din cele 3 numere poate fi folosit de zero sau mai multe ori.

Cerință

Determinați numărul minim de cifre pe care îl poate avea un număr având proprietățile precizate mai sus.

Date de intrare

Fișierul de intrare **base3.in** conține 3 linii. Pe fiecare linie se află scris un număr în baza 3.

Date de ieșire

Fișierul de ieșire **base3.out** va conține numărul minim de cifre pe care îl poate avea un număr N cu proprietățile specificate. Dacă nu se poate obține nici un astfel de număr, afișați în fișier valoarea 0.

Restricții și precizări

- Numărul de cifre al fiecăruia din cele 3 numere este un număr întreg între 1 și 16000.
- Numerele date pot conține zerouri la început; acestea trebuie luate în considerare, dacă numărul respectiv este folosit în concatenare.

Exemplu

base3.in	base3.out
001	13
020	
2020	

Explicație:

Se poate obține numărul *2020001001001* .

Timp maxim de executare/test: 0.4 secunde pentru Linux și 0.8 secunde pentru Windows