

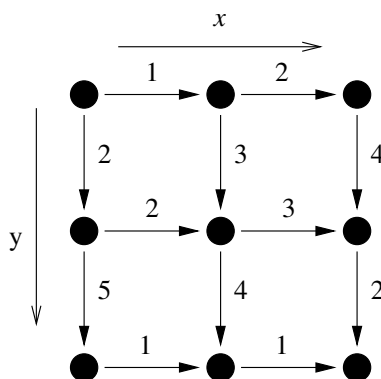
Vhodna datoteka: -  
Izhodna datoteka: -  
Izvorna koda: square.pas / .c / .cpp

100 točk  
Časovna omejitev: 1 s  
Pomnilniška omejitev: 16 MB

## Square

Dan je graf, kjer so točke razporejene v obliki kvadratne mreže velikosti  $N \times N$  ( $1 \leq N \leq 2003$ ). Iz vsake točke vodijo povezave do njene desne in spodnje sosede, če take sosede seveda obstajajo (na desnem in spodnjem robu jih naprimer ni). Povezave med točkami so utežene s celoštevilskimi utežmi  $w$  ( $1 \leq w \leq 500\,000$ ). Če medsebojno spojimo eno ali več zaporednih povezav dobimo pot. Tudi pot ima svojo utež, ki je enaka vsoti uteži vseh povezav, ki ležijo na tej poti. Uteži so razporejene tako, da imajo vse možne poti iz izhodiščne točke  $v_{1,1}$  levo zgoraj do neke poljubne točke  $v_{x,y}$  enako utež.

Na spodnjem primeru je podan graf velikosti  $N = 3$ . Vse poti iz izhodišča  $v_{1,1}$  do točke  $v_{2,2}$  imajo utež 4. Edina pot od  $v_{1,1}$  do  $v_{1,2}$  ima utež 2. Podobno velja za ostale točke.



Dano je celo število  $L$  ( $1 \leq L \leq 2\,000\,000\,000$ ). Vaša naloga je poiskati točko  $v_{x,y}$ , tako da bodo imele poti od izhodiščne točke  $v_{1,1}$  do izbrane točke  $v_{x,y}$  utež natanko  $L$ . Uteži posameznih povezav programu niso vnaprej podane. Namesto tega je programu na voljo dodatna knjižnica, ki jo mora program povprašati za utež za vsako povezavo posebej. Program sme povprašati za uteži največ 6667-krat.

Knjižnica vsebuje naslednje funkcije: `getN()` in `getL()`, ki vrmeta števili  $N$  in  $L$ , `getWeight(x, y, direction)`, ki vrne utež za povezavo, ki povezuje točko  $v_{x,y}$  s svojo desno (`direction=0`), oz. spodnjo (`direction=1`) sosedo.

Če obstaja takšna točka  $v_{x,y}$ , da ima pot od  $v_{1,1}$  do  $v_{x,y}$  utež natanko enako  $L$ , potem naj program pokliče podprogram `solution(x, y)`. Če takšna točka ne obstaja, potem naj program pokliče `solution(-1, -1)`. Po klicu podprograma `solution` bo program avtomatično prekinjen. Če program več kot 6667-krat pokliče `getWeight()`, potem se program napačno izvaja in ne dobi nobene točke za dan vhodni primer.



# CENTRAL EUROPEAN OLYMPIAD IN INFORMATICS

Münster, Germany  
5-12 julij, 2003

Stran 2 od 2

Slovenia

Dan 1: **square**

Naloga nima nobene vhodne oz. izhodne datoteke.

## Funkcije knjižnice

C/C++
<pre>int getN(void) int getL(void) int getWeight(int x, int y, int direction) void solution(int x, int y)</pre>
Pascal
<pre>function getN: Longint function getL: Longint function getWeight(x, y, direction: Longint): Longint procedure solution(x, y: Longint)</pre>

V mapi `~/ceoi` oz. `c:\ceoi` lahko najdete primer knjižnice `square.lib`, vendar upoštevajte, da bo program med ocenjevanjem preveden z neko drugo verzijo knjižnice.

Za razvoj in testiranje programa lahko sami napišete datoteko `square.in`, ki jo bo uporabila priložena knjižnica. Prva vrstica datoteke `square.in` naj vsebuje števili  $N$  in  $L$ . Naslednjih  $N$  vrstic vsebuje natanko  $N - 1$  celih števil, ki predstavljajo uteži na vodoravnih povezavah. Naslednjih  $N - 1$  vrstic pa vsebuje  $N$  celih števil, ki predstavljajo uteži na navpičnih povezavah.

Knjižnica naj bo vključena z `#include "square.lib.h"` oz. `uses square.lib`.

Upoštevajte, da je priložena knjižnica precej počasna in obenem porabi precej pomnilnika. To je zato, ker mora vhodno datoteko naložiti v pomnilnik. Za evaluacijsko verzijo knjižnice, s katero bo program preveden med ocenjevanjem, lahko privzamete, da ne bo uporabljala nič pomnilnika in ne bo porabila nič časa, ki je na voljo vašemu programu.

## Primer vhodne datoteke za uporabo testne knjižnice

square.in	Protocol
3 4	<code>getN() -&gt; 3</code>
1 2	<code>getL() -&gt; 4</code>
2 3	<code>getWeight(1,1,0) -&gt; 1</code>
1 1	<code>getWeight(2,1,1) -&gt; 3</code>
2 3 4	<code>solution(2,2)</code>
5 4 2	