4. Српска информатичка олимпијада
II дан
Београд – 30. мај 2010.

РОБОТИ

Омиљене играчке Огњена и Вука били су роботи. Једног дана су их све сакупили, на бетону исцртали велику квадратну мрежу на чија поља су сместили роботе и пустили их да се крећу. С обзиром да их је било пуно и да нису имали сензоре који би их упозорили да треба да промене правац кретања, роботи су се често сударали.
Ваш задатак је да напишете програм **ROBOTI** којим би се одредио смер кретања сваког од робота, али тако да се никоја два не сударе, при чему:

- се сваки робот може кретати по квадратној мрежи само по хоризонтали и вертикали, дакле у једном од четири смера: лево, десно, горе и доле;
- позиција сваког робота је описана његовим координатама, односно хоризонталном и вертикалном удаљеношћу (цели бројеви), у односу на доњи леви угао мреже (дакле, поље које се налази у доњем левом углу има координате 1 и 1, а прво десно од њега 2 и 1);
- постоји више начина како се могу усмерити роботи, а ви ћете поштовати следећи план усмеравања робота: ако је могуће робота треба усмерити лево, ако није онда десно, ако није ни то могуће онда горе и на крају, ако је све претходно немогуће онда га усмерити на доле.

Улазни подаци. Стандардни улаз садржи више линија:

- прва садржи један позитиван цео број робота N (N ≤ 1 000),
- наредних N линија садржи по два цела броја, где први представља хоризонталну, а други вертикалну удаљеност текућег робота од доњег левог угла мреже.

Излазни подаци. Стандардни излаз садржи -1 ако није могуће избећи сударање робота, а ако је могуће оријентисати их тако да се никоја два не сударају тада излаз садржи низ од N карактера, где је првим карактером описан смер првог робота, другим другог итд. и то тако да карактер L одговара кретању у лево, R у десно, U на горе и D на доле.

Пример.

Улаз:	Излаз:
3 1 1 3 3 2 2	LLL

Улаз:	Излаз:
5 1 2 2 2 2 1 2 3 3 2	-1

DOWNLOAD

За подухват издраде кућице на дрвету Огњен је решио да се, за сваки случај, опреми најразличитијим Уради сам књигама, док је Вук тражио неке бесплатне софтвере за визуелизацију осмишљених објеката. Тражећи по Интернету, скидали су с реда све што им се учинило да ће им можда затребати. Обзиром да им је проток био прилично ограничен закључили су да ће download потрајати, као и да није тако тешко одредити колико.
Ваш задатак је да напишете програм **LOAD** који ће одредити колико ће секунди трајати download ако се зна да је укупни могући проток P kB непроменљив, да се фајлови скидају паралелно и да се у тренутку када се заврши download једног фајла његов проток расподељује осталима и то тако да се минимизује време за које ће се завршити пренос свих фајлова.

Улазни подаци. У првој линији улаза се налазе два цела броја P, који представља максималан проток, и N који преставља број фајлова који се download-ују (P≤ 1 000,N ≤ 1 000). Свака од наредних N линија садржи по два цела броја, где први S представља број секунди који је потребан за download фајла (на који се та линија односи) при брзини преноса B (kB/s) која је задата другим бројем из исте линије (B ≤ 1 000).

Пример.

Улаз:	Улаз:
4 3 1 10 1 20 2 40 Излаз: 27.5	40 5 1 10 2 10 1 40 2 20 1 50 Излаз: 4.0

Излазни подаци. Излаз садржи један позитиван реалан број који представља број секунди потребан за скидање свих фајлова.

РЕСТОРАНИ

У својим маштањима, Огњен и Вук су често правили планове о томе како ће зарадити новац. Један од њихових идеја било је и отварање ланца ресторана домаће хране у булевару где су се налазили високи солитери са великим бројем станара. Није им било тешко да проматрају и анкетирају, па су захваљујући томе сазнали колика је пожељна удаљеност ресторана, тј. максимална удаљеност ресторана коју су станари солитера спремни да пређу да би ручали у ресторану, а прескочили спремање ручка код куће. Када су то утврдили и договорили колико би ресторана било добро отворити за почетак, утврдили су да постоји више локација за отварање ресторана него што би њима било потребно. Остало је још одабрати локације тако да ресторани буду на пожељној удаљености од што већег броја станара солитера.

Ваш задатак је да напишете програм **RESTORAN** који би могао да обави посао уместо Огњена и Вука, тј. којим би се:

- за дату пожељну удаљеност од солитера R($R \leq 2\,000\,000$), број ресторана K ($K \leq 30$)
- број солитера N ($N \leq 30$), удаљеност сваког солитера од почетка улице
- удаљеност потенцијалних локација за ресторане од почетка улице (њих такође има N), која није већа од 2 000 000

одредило које локације треба одабрати, да би задовољиле претходно описан критеријум. Напомена: Сви солитери имају исти број становника.

Улазни подаци. Прва линија стандардног улаза садржи три цела броја: R (пожељна удаљеност ресторана од солитера), K (број ресторана које треба отворити) и N (број зграда, који је уједно и број локација). Наредних N линија садржи по један позитиван цео број који представља удаљеност једног солитера од почетка булевара, следећих N линија садржи по један позитиван цео број који представља удаљеност једне локације од почетка булевара. И за солитере и за локације дат је растући улаз, тј. први солитер/локација је најближи почетку, за њим други итд.

Излазни подаци. Стандардни излаз садржи K линија са по једним целим бројем, који престаља редни број изабране локације.

Напомена: Уколико постоји неколико решења, бирати оно у којем је прва (она са најмањим редним бројем) ближа почетку улице.

Пример.

Улаз:	Излаз:	Коментар на пример:
4 1 3 2 6 12 1 8 11	2	Ако је ресторан на 2. локацији, онда ће бити на довољној удаљености од 2 солитера, а остале локације одговарају станарима само једног солитера.

Улаз:	Излаз:	Коментар на пример:
2 2 3 1 4 7 2 5 6	1 2	Локације (1,2) и (1,3) имају у свом простору 3 зграде. Бирамо прву.