

BUC	CRA	DOB	EFO	FAG	GIU	HIR	IAS	LUG	MEH	NEA	ORA	PIT	RIM	SIB	TIM	URZ	VAS	ZER	
418	366	374	687	239	508	601	737	229	299	824	291	317	220	140	118	503	645	75	ARAD
	239	359	269	211	90	183	319	504	434	406	429	101	198	278	615	85	227	500	BUCHAREST
		120	508	325	329	422	558	265	195	645	377	138	146	226	376	324	466	448	CRAIOVA
			628	445	449	542	678	145	75	765	497	258	266	346	256	444	586	568	DOBRETA
				480	359	86	418	773	703	505	698	370	467	547	884	184	326	769	EFORIE
					301	394	530	590	520	617	250	276	179	99	701	296	438	321	FAGARAS
						273	409	594	524	496	519	191	288	368	705	175	317	590	GIURGIU
							332	687	617	419	612	284	381	461	798	98	240	683	HIRSOVA
								823	753	87	748	420	517	597	934	234	92	819	IASI
									70	910	642	403	411	491	111	589	731	713	LUGOJ
											840	572	333	341	421	181	519	661	MEHADIA
												835	507	604	684	1021	321	179	906
													328	231	151	753	514	656	71
														97	177	514	186	328	399
															80	522	283	425	302
																602	363	505	222
																	700	842	824
																		142	585
																			727
																			824
																			URZICENI
																			VASLUI
																			ZERIND

Taulukko 1: Romanian pääteiden välimatkataulukko. (Vaakarivillä kaupunkien nimet on lyhennetty.)

Datatähti 2003 -kilpailu

1 Välimatkataulukko

Olet autoilulomalla Romaniassa. Olet tällä hetkellä ARADin kaupungissa, ja sinun pitäisi päästä BUCHARESTin kaupunkiin. Ajoreitin suunnittelua haittaa kuitenkin se, ettei sinulla ole Romanian tiekarttaa. Onneksi sinulla on edes välimatkataulukko 1, joka kertoo lyhyimmät etäisyydet Romanian suurimpien kaupunkien välillä pääteitä pitkin seuraavasti: pystysarakkeelta valitaan aakkosjärjestyksessä ensimmäinen ja vaakariviltä toinen kaupunki, ja niiden leikkauskohdasta voi lukea vastaavan etäisyyden. Taulukosta voit siis lukea, että ajomatkaa on 418 km. Miten selvität taulukosta, minkä taulukossa mainittujen kaupunkien kautta tuo lyhyin reitti kulkee?

1.1 Tehtävä

Toteuta sellainen ohjelma, joka saa syötteenään kaupunkien lukumäärän ja näiden kaupunkien väliset minimietäisyydet, ja tulostaa järjestyksessä ne kaupungit, joiden kautta varrella ensimmäisen ja toisen syötekaupungin välinen lyhyin reitti kulkee. Jos tällaisia lyhyimpiä reittejä on useita, voi ohjelmasi tulostaa niistä minkä tahtoo.

1.2 Syöttö

Ohjelmasi lukee oletussyöttöä (standard input). Ensimmäisellä rivillä on kaupunkien lukumäärä $3 \leq n \leq 1\,000$. Seuraavalta riviltä alkaa välimatkataulukko: Ensimmäiset $n-1$ lukua kertovat kaupungin 1 minimietäisyydet muihin kaupunkeihin $2, 3, 4, \dots, n$ tässä järjestyksessä. Seuraavat $n-2$ lukua kertovat kaupungin 2 etäisyydet jäljellä oleviin kaupunkeihin $3, 4, 5, \dots, n$ ja niin edelleen, kunnes kaikki etäisyydet on annettu. Kukin etäisyys on kokonaisluku väliltä $1 - 20\,000$ (km). Etäisyyksien välissä on joko välilyönti tai rivinvaihto.

Esimerkiksi taulukossa 1 kaupunki on numeroitu aakkosjärjestyksessä, joten vastaava syöte koostuu seuraavista numeroista:

418 366 374 687 239 508 601 737 229 299 824 291 317 220 140 118 503 645 75
239 359 269 211 90 183 319 504 434 406 429 101 198 278 615 85 227 500 120
508 325 329 422 558 265 195 645 377 138 146 226 376 324 466 448 628 445 449
542 678 145 75 765 497 258 266 346 256 444 586 568 480 359 86 418 773 703
505 698 370 467 547 884 184 326 769 301 394 530 590 520 617 250 276 179 99
701 296 438 321 273 409 594 524 496 519 191 288 368 705 175 317 590 332 687
617 419 612 284 381 461 798 98 240 683 823 753 87 748 420 517 597 934 234
92 819 70 910 642 403 411 491 111 589 731 713 840 572 333 341 421 181 519
661 643 835 507 604 684 1021 321 179 906 328 231 151 753 514 656 71 97 177
514 186 328 399 80 522 283 425 302 602 363 505 222 700 842 824 142 585 727

1.3 Tulostus

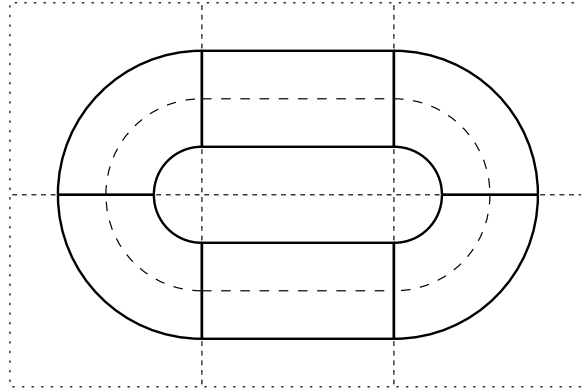
Ohjelmasi tulostaa oletustulostukseen (standard output). Vastauksena tulostetaan jokin minimireitti kaupungista 1 kaupunkiin 2 sellaisena jonona kaupunkien numeroita, että

- ensimmäinen on 1
- viimeinen on 2
- kahden peräkkäisen kaupungin välisellä minimireitillä ei ole muita syötteen kaupunkeja
- näin kuljettu kokonaismatka on kaupunkien 1 ja 2 välinen minimietäisyys.

Tämä jono tulostetaan luettelemalla sen numerot järjestyksessä ensimmäisestä viimeiseen kukin omana rivillään.

Esimerkiksi taulukon 1 ensimmäinen eli lähtökaupunki on ARAD ja toinen eli kohdekaupunki on BUCHAREST, ja niiden välinen lyhyin reitti kulkee välikaupunkien SIBIU, RIMNICU-VILCEA ja PITESTI kautta. Vastaava tulostus on siis:

1
16
15
14
2



Kuva 1: Teuvon ovaalirata

Datatähti 2003 -kilpailu

2 Autorata

Teuvo on saanut joululahjaksi autoradan. Se on kuvan 1 mukainen klassinen ovaalirata joka koostuu neljästä kaarre- ja kahdesta suorasta palasta. Tällaisia paloja voi käännellä ja yhdistellä vapaasti, kunhan tuloksena on suljettu kaksikaistainen ratasilmukka. Silmukka ei myöskään voi ylittää itseään, koska radassa ei ole siltojen tekemiseen tarvittavia osia.

Samantyyppisen ovaaliradan ainainen tahkoaminen alkaa kuitenkin kyllästyttää jopa Teuvoa, joten hän hakee kierrätyskeskuksesta autorataansa uusia paloja. Nyt hän haluaa koota kaikista palloistaan yhden tosi ison autoradan, mutta tarvitsee sen suunnitteluun apua.

2.1 Tehtävä

Toteuta sellainen ohjelma, jolle annetaan syötteenä montako uutta palaa Teuvo löysi, ja joka tulostaa jonkin sellaisen radan, jonka Teuvo pystyy nyt rakentamaan. Radasta saa jättää pois korkeintaan yhden kaarre- ja yhden suoran palan, mutta kaikki muut palat on käytettävä.

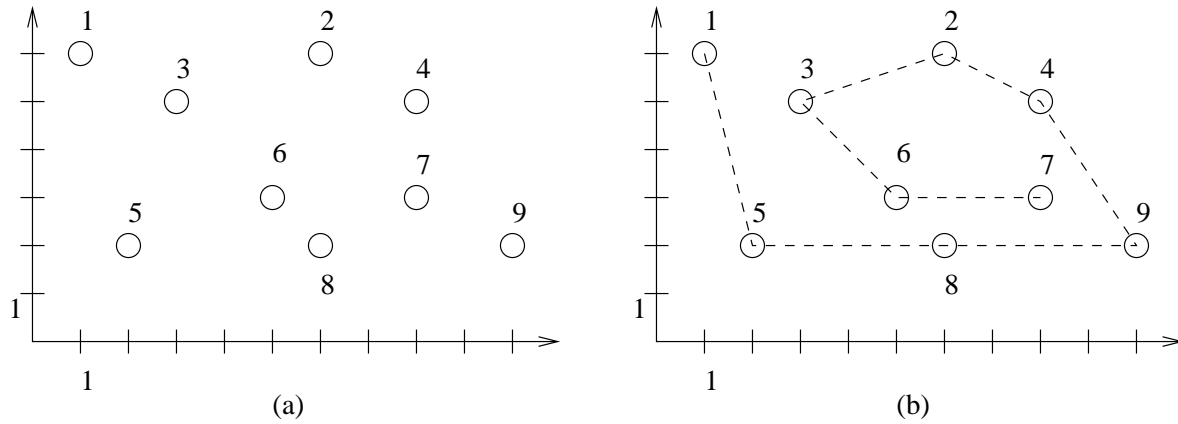
Jos esimerkiksi Teuvo löysi kuusi uutta kaarrepalaa ja yhden uuden suoran palan, niin hän voisi nyt rakentaa esimerkiksi kuvan 2 mukaisen radan, jossa Teuvolta jäi käyttämättä yksi kolmesta suorasta palasta.

2.2 Syöttö

Ohjelmasi lukee oletussyöttöä (standard input). Ensimmäisellä rivillä on uusien kaarrepalojen lukumäärä m . Toisella rivillä on uusien suorapalojen lukumäärä n . Molemmat luvut ovat kokonaislukuja väliltä 1 – 1 000 000.

2.3 Tulostus

Ohjelmasi tulostaa oletustulostukseen (standard output) radan kuvauksen seuraavasti: Aloitetaan radan jostakin palasta radan kiertäminen vastapäivään. Joka palan kohdalla tulostetaan, onko se *suora* eli *s*, *vasen* kaarre eli *v*, vaiko *oikea* kaarre eli *o*. Sitten siirrytään kiertosuunnassa



Kuva 3: Pistejoukko ja siihen piirretty murtoviivaspiraali

Datatähti 2003 -kilpailu

3 Spiraali

Tässä ongelmassa annetaan joukko pisteitä tasossa, ja pyydetään piirtämään niiden kaikkien kautta kulkeva murtoviiva, joka kiertyy itsensä ympäri kuin spiraali. Kuva 3 (a) näyttää erään annetun pistejoukon, kuva 3 (b) taas erään sen kautta piirretyn murtoviivaspiraalin.

3.1 Tehtävä

Toteuta sellainen ohjelma, jolle annetaan syötteenä n pistettä tasossa, ja joka tulostaa saamansa pisteet sellaisessa järjestyksessä $p_1 p_2, p_3, \dots, p_n$, että seuraava spiraaliehto pätee:

Otetaan mitkä tahansa tulosteen kolme peräkkäistä pistettä p_i, p_{i+1}, p_{i+2} . Kun ensimmäisestä pisteestä p_i katsotaan kolmannen pisteen p_{i+2} suuntaan, niin väliin jäävä toinen piste p_{i+1} ei jää näkökentän vasemmalle puoliskolle, vaan on joko suoraan näkökentän keskellä tai oikealle puoliskolla. Lisäksi kaikki muut näkökentän oikealle puoliskolle jäävät pisteet on tulostettu jo aiemmin, eli kuuluvat pisteisiin p_1, \dots, p_{i-1} .

Kuvassa 3 (b) tämä spiraaliehto toteutuu kun pisteet tulostetaan vastapäivään kiertäen.

3.2 Syöttö

Ohjelmasi lukee oletussyöttöä (standard input). Ensimmäisellä rivillä on pisteiden määrä $3 \leq n \leq 1\,000$. Kullakin seuraavalla n rivillä on yhden pisteen paikka seuraavasti: ensin tulee sen x -koordinaatti, sitten väli, ja sitten sen y -koordinaatti. Kumpikin koordinaatti on kokonaisluku väliltä $0, \dots, 1\,000\,000$. Voit olettaa, ettei mikään piste toistu syötössä.

Jos esimerkiksi kuvan 3 (a) pisteet annetaan syötteenä siihen kuvaan merkityssä numerojärjestyksessä, niin tämän <syötteen ulkoasu on seuraava:

9
1 6
6 6
3 5
8 5
2 2
5 3
8 3
6 2
10 2

3.3 Tulostus

Ohjelmasi tulostaa oletustulostukseen (standard output). Tulostus koostuu n rivistä siten, että rivillä r on vain se järjestysnumero $1 \leq j \leq n$, joka kertoo, että tulostusjärjestyksessä r . pisteenä on syötejärjestyksen j . piste.

Esimerkiksi kuvaa 3 (b) vastaava tulostus on seuraava:

1
5
8
9
4
2
3
6
7

Ratkaisujen palautus

Kunkin tehtävän ratkaisuna sinun pitää palauttaa asianmukaisesti kommentoitu lähdekielinen ohjelma omana tiedostonaan. Koko vastauksesi koostuu siis kolmesta tiedostosta. Niiden nimet ovat muotoa `dt2003rn.k` seuraavasti:

- Indeks $n = 1, 2, 3$ on vastaavan tehtävän numero tämän dokumentin mukaisesti.
- Pääte k on tiedostossa käytetyn ohjelmointikielen mukaisesti yksi seuraavista:

k	kieli
<code>pas</code>	Pascal
<code>c</code>	C
<code>cpp</code>	C++

Älä käytä tiedostoissasi yli 80 merkin rivejä!

Sallittuja ohjelmointikieliä ovat siis Pascal, C ja C++. Laita ohjelmasi alkuun kommentiksi yhteystietosi ja mahdolliset ohjeet ohjelman kääntämistä varten. Ohjelmalla pitää olla vain edellä määritellyt syötet ja tulosteet. Erityisesti sen ei pidä lukea mitään näppäimistöltä tai tulostaa mitään näyttöruudulle.

Arvosteluvaiheessa ohjelmasi käännetään (Linux-ympäristössä) ja suoritetaan useilla erilaisilla testisyötteillä. Pisteitä saa käännetyn ohjelman oikeasta ja nopeasta toiminnasta joukolla testisyötteitä. Lähdekielistä ohjelmaa ei siis arvostella, kunhan vain sen käännös onnistuu. Siirrettävyysongelmien välttämiseksi pitäydy valitsemasi kielen peruspiirteissä.

Testausympäristössä on kääntäjinä käytettävissä GNU C/C++ <http://gcc.gnu.org/> ja Free Pascal <http://www.freepascal.org/>, jonka pitäisi olla Turbo Pascal -yhteensopiva. Käännökset tehdään kääntäjien oletusarvoisilla valitsimilla.

Testisyötteistä osa testaa vain ohjelman oikeellisuutta. Kustakin tällaisesta syötteestä saa tietyn määrän pisteitä, jos ohjelma tuottaa oikean tulosteen kohtuullisessa ajassa. Syötteiden koot ja aikarajat valitaan siten, että hitaallakin ohjelmalla saa pisteitä, kunhan se on muuten toimiva. Osa testisyötteistä testaa ohjelman tehokkuutta. Näillä syötteillä ohjelman saama pistemäärä vähenee sitä mukaa, kun suoritus aika kasvaa. Testisyötteet ovat aina tarkalleen ylläolevan kuvauksen mukaisia. Pisteytyksen yksityiskohtia ja tarkkoja aikarajoja ei tässä vaiheessa julkisteta.

Palauta ratkaisusi sähköpostitse osoitteeseen matti.nykanen@cs.helsinki.fi. Laadi viesti, jonka otsikossa mainitaan "Datatähti 2003", ja joka sisältää yhteystietosi. Liitä sitten jokainen yllämainituista kolmesta lähdekooditiedostostasi omana MIME-liitteenään tähän viestiisi. Viestissäsi tulisi olla siis kolme eri liitettä.

Jos MIME-liitteiden käyttö on mahdotonta, niin lähetä ensin yllä kuvattu viesti, jossa kerrot myös, että lähetät tiedostosi erikseen. Lähetä sen perään kukin kolmesta lähdekooditiedostostasi omana viestinään.

Voit myös toimittaa ratkaisusi 3,5 tuuman DOS-formatoidulla levykkeellä osoitteeseen

Matti Nykänen
Datatähti 2003 -alkukilpailu
Tietojenkäsittelytieteen laitos
PL 26 (Teollisuuskatu 23)
00014 Helsingin yliopisto

Pakkaa levyke huolellisesti ja laita mielellään toinen samansisältöinen mukaan varmuuskopioksi. Talleta yllämainitut kolme lähdekooditiedostoasi levykkeesi juurihakemistoon.

Jos sinulla on kysymyksiä näistä tehtävistä tai niiden palauttamisesta, niin kysy asiasta sähköpostitse tai kirjeitse edellä mainittuihin osoitteisiin.