

Upute natjecateljima

Ovo natjecanje sastoji se od rješavanja **6 problemskih zadataka** u vremenu od **3 sata** (180 minuta). Zadatke je potrebno rješavati u jednom od sljedećih programskih jezika: Pascal, C ili C++. Natjecatelji u Pascalu moraju programirati u programskom alatu **Free Pascal**, a natjecatelji u C-u i C++-u programskom alatu **DJGPP**. Dozvoljeno je koristiti editor po izboru te pomoću navedenih alata prevoditi izvorni kod u izvršnu datoteku.

Za početak, provjerite da ste dobili od vašeg voditelja **točno 11 stranica** (uključujući ovu koju trenutno čitate). Morate imati dvije stranice s uputama (ovu i još jednu), jednu uvodnu stranicu s tablicom zadataka, i još 8 stranica s tekstovima zadataka.

Tijekom natjecanja **ne smijete komunicirati** ni sa jednom osobom, osim voditelja natjecanja. To znači da morate **raditi samostalno i ne smijete koristiti Internet**. Također, zabranjena je upotreba bilo kakvih ranije napisanih programa ili dijelova programa.

Po isteku vremena predviđenog za natjecanje, morate voditelju natjecanja predati disketu sa snimljenim izvornim kôdovima rješenja. Nakon natjecanja, sustav za evaluaciju će testirati vaša rješenja na, unaprijed zadanim, test podacima i dodijeliti vam određeni broj bodova. Na kraju svakog zadatka dani su primjeri test podataka. Ti primjeri služe da bi vam tekst zadataka bio što je moguće jasniji te za provjeru formata ulaza i izlaza, a ne služe za provjeru ispravnosti vašeg programa. Ako vaš program radi na tim primjerima, to **nije garancija** da će raditi za ijedan službeni test podatak.

Pripazite na to da ne nosi svaki zadatak jednak broj bodova. Lakše i brže rješivi zadaci nose manje bodova, dok teži nose više bodova. Svaki test podatak u nekom zadatku nosi jednak broj bodova. Ukupni broj bodova na nekom zadatku je jednak zbroju bodova test podataka koji se poklapaju sa službenim rješenjem. Ukupan broj bodova jednak je zbroju bodova na svim zadacima.

Primjetite da **svi zadaci imaju ulaz sa standardnog ulaza (tipkovnice) i izlaz na standardni izlaz (monitor)**. Sve informacije o zadacima (ime zadatka, ulaz, izlaz, vremensko i memorijsko ograničenje, način bodovanja) možete naći na uvodnoj stranici s naslovom *Zadaci*. Ako vam nije jasno nešto u vezi načina provedbe ovog natjecanja, odmah postavite pitanje voditelju da vam to razjasni.

Tijekom cijelog natjecanja možete postavljati pitanja voditelju natjecanja u vezi zadataka. Dozvoljena su pitanja **koja razjašnjavaju nejasnoće u tekstu zadatka**. Ne smijete postavljati pitanja u vezi rješavanja zadataka. Prije nego postavite pitanje, pročitajte još jednom zadatak, jer je moguće da ste u prethodnom čitanju preskočili dio teksta zadatka.

Upute natjecateljima

VAŽNO za C/C++!

Glavni program (glavna funkcija) **mora** biti deklariran kao: `int main(void) { ... }`.

Program mora završiti svoje izvođenje naredbom `return 0;` unutar funkcije `main` ili naredbom `exit(0);`.

Zabranjeno je koristiti biblioteke `<conio.h>` i `<cconio>`, kao i sve funkcije deklarirane u ovim bibliotekama (npr. `clrscr()`; `getch()`; `getche()`; i sl.). Zabranjeno je koristiti i **sve** sistemske (nestandardne) biblioteke.

Zabranjeno je koristiti funkcije `itoa()` i `ltoa()` jer one ne postoje u standardu jezika C/C++. Umjesto tih funkcija možete koristiti funkciju `sprintf()`, koja ima i veće mogućnosti primjene, deklariranu u `<stdio.h>` i `<cstdio>`.

Zabranjeno je koristiti sve funkcije za rad s datotekama (`fopen()`, `fclose()`, `fprintf()`, `fscanf()`, ...), kao i deklarirati varijable tipa `FILE *`.

Dozvoljeno je koristiti sve ostale standardne biblioteke (koje su dio jezika), uključujući i STL (Standard Template Library) u jeziku C++.

VAŽNO za Pascal!

Program **mora** regularno završiti svoje izvođenje naredbom `end.` unutar glavnog programa ili naredbom `halt;`.

Zabranjeno je koristiti bilo kakve biblioteke, a posebno biblioteku `crt`, tj. zabranjeno je u programu imati direktivu `uses`. To znači da u programu ne smije biti naredbi `clrscr()` i `readkey()`.

Nepoštivanje ovih uputa ili nepridržavanjem formata izlaznih podataka rezultirat će nepovratnim gubitkom bodova. Nemojte ispisivati ništa što se u zadatku ne traži, kao npr. poruke tipa 'Rjesenje je:' ili 'Unesite brojeve' i slično!

Napomena: ista ovakva ograničenja će biti i na ovogodišnjem Državnom natjecanju.

Nakon natjecanja, sva eventualna pitanja, komentare i mišljenja o zadacima uputite na donju e-mail adresu.

Sretno i uspješno natjecanje!

Autor zadataka, test podataka i rješenja

Luka Kalinovčić, kalinovcic@gmail.com

Zadaci

Zadatak	PATULJCI	NPuzzle	TROJKE	TENKICI	BICIKLI	LISTA
Izvorni kôd	patuljci.bas patuljci.pas patuljci.c patuljci.cpp	npuzzle.bas npuzzle.pas npuzzle.c npuzzle.cpp	trojke.bas trojke.pas trojke.c trojke.cpp	tenkici.bas tenkici.pas tenkici.c tenkici.cpp	bicikli.bas bicikli.pas bicikli.c bicikli.cpp	lista.bas lista.pas lista.c lista.cpp
Ulazna datoteka	standardni ulaz (tipkovnica)					
Izlazna datoteka	standardni izlaz (zaslon)					
Memorijsko ograničenje (heap)	32 MB	32 MB	32 MB	32 MB	32 MB	32 MB
Memorijsko ograničenje (stack)	8 MB	8 MB	8 MB	8 MB	8 MB	8 MB
Vremensko ograničenje (po test podatku)	1 sekunda	1 sekunda	1 sekunda	1 sekunda	1 sekunda	1 sekunda
Broj test podataka	10	10	10	14	10	9
Broj bodova (po test podatku)	2	2	3	5	7	10
Ukupno bodova	20	20	30	70	70	90
	300					

Napomena: Vremensko ograničenje je mjereno na računalu baziranom na dva procesora AMD Athlon MP 2600+ i operativnim sustavom Linux. Program u C-u i C++-u treba kompajlirati sa sljedećim opcijama: `-O2 -lm -static`, a program u Pascalu s `-O1 -XS`. U programskom jeziku Qbasic dozvoljeno vrijeme izvršavanja programa je dvostruko veće.

1. PATULJCI

Svaki dan dok patuljci rade u rudniku Snjeguljica posprema kuću i kuha ručak. 7 stolica, 7 tanjurića, 7 vilica i 7 noževa za 7 gladnih patuljaka.

Međutim, jednoga se dana iz rudnika vratilo 9 patuljaka (nitko ne zna kako ni zašto) i svaki od njih tvrdi da je on jedan od njenih 7 patuljaka.

Srećom, svaki patuljak na svojoj kapici ima jedinstven prirodni broj manji od 100, a Snjeguljica matematičarka, još je davno primjetila da zbroj brojeva na kamicama njenih 7 patuljaka iznosi točno 100.

Napišite program koji će prepoznati koji su patuljci pravi, odnosno odabrati 7 od 9 brojeva čiji zbroj iznosi 100.

Ulazni podaci

U devet redaka nalazi se po jedan prirodni broj između 1 i 99. Brojevi će biti međusobno različiti.

Napomena: Ulazni podaci bit će takvi da će rješenje biti jedinstveno.

Izlazni podaci

U sedam redaka treba ispisati po jedan prirodni broj – to su brojevi na kamicama Snjeguljičinih sedam patuljaka. Brojeve možete ispisati bilo kojim redom.

Primjeri test podataka

Ulaz	ulaz
7	8
8	6
10	5
13	1
15	37
19	30
20	28
23	22
25	36
izlaz	izlaz
7	8
8	6
10	5
13	1
19	30
20	28
23	22

2. NPUZZLE

N-puzzle je slagalica poznata pod mnogim imenima i u mnogim varijantama. U ovom zadatku promatrat ćemo tzv. 15-puzzle. Ona se sastoji od $4 * 4$ mreže klizećih kvadratića gdje jedan kvadratić nedostaje. Kvadratići su označeni slovima od 'A' do 'O', a složena slagalica izgleda ovako:

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	.

Ponekad je korisno (npr. kod rješavanja uz pomoć računala) definirati “razbicanost” slagalice kao zbroj udaljenosti svakog klizećeg kvadratića do njegovog konačne pozicije. Udaljenost između dvije pozicije je apsolutna razlika redaka zbrojena s apsolutnom razlikom stupaca.

Napišite program koji će izračunati razbicanost zadane slagalice.

Ulazni podaci

U četiri retka nalazi se po četiri znaka koji predstavljaju stanje slagalice.

Izlazni podaci

U prvi i jedini redak ispišite razbicanost slagalice.

Primjeri test podataka

ulaz

ABCD
EFGH
IJKL
M.NO

izlaz

2

ulaz

.BCD
EAGH
IJFL
MNOK

izlaz

6

3. TROJKE

Mirko i Slavko igraju novu igru – “Trojke”. Prvo kredom na cesti nacrtaju $N \times N$ kvadratnu mrežu. Zatim u neka polja upišu slova. Nijedno se slovo **ne nalazi** na više od jednog mjesta u mreži.

Nakon toga, potrebno je što brže otkriti tri slova koja leže na istom pravcu. Tri slova leže na istom pravcu ako postoji pravac koji prolazi središtem **svakog** od tri kvadratića.

Nakon nekog vremena postaje sve teže pronaći novu trojku. Zbog toga im treba program koji će im reći koliko ukupno ima trojki, kako bi znali jesu li pronašli sve ili trebaju još tražiti.

Ulazni podaci

U prvom redu nalazi se prirodni broj N ($3 \leq N \leq 100$) – dimenzija mreže.

U sljedećih N redaka nalazi se po N znakova koji opisuju mrežu. Znak '.' označava prazno polje, dok veliko slovo engleske abecede označava slovo u kvadratiću.

Izlazni podaci

U prvi i jedini redak ispišite ukupni broj trojki.

Primjeri test podataka

Ulaz

4
...D
..C.
.B..
A...

izlaz

4

ulaz

5
..T..
A....
.FE.R
....X
S....

izlaz

3

Ulaz

10
....AB....
..C....D..
.E.....F.
...G..H...
I.....J
K.....L
...M..N...
.O.....P.
..Q....R..
....ST....

izlaz

0

4. TENKICI

Mirko je na djedovom tavanu pronašao kolekciju od N tenkića iz drugog svjetskog rata. Odmah je pozvao Slavka da se zajedno igraju. Napravili su bojišnicu – drvenu ploču podijeljenu na $N * N$ kvadratića.

U njihovoj igri tenk se može u jednom potezu pomaknuti na jedno od četiri susjednih polja i taj pomak nazivamo **korakom**. Tenk može pucati na bilo koje polje unutar istog retka i unutar istog stupca. Kažemo da tenk **čuva** redak i stupac u kojem se nalazi.

Dodatno, ni u kojem trenutku na istom polju **ne smiju biti** dva ili više tenka.

Nakon više sati igranja, Mirkova mama ih je pozvala na ručak, a oni su odlučili postaviti tenkove tako svaki tenk čuva jedan stupac i jedan redak, odnosno da se u svakom retku i svakom stupcu nalazi točno jedan tenk.

To ne bi bio nikakav problem da to nisu odlučili napraviti u minimalnom broju koraka.

Napišite program koji će pronaći minimalni broj koraka potreban za postaviti tenkove svaki u svoj redak i stupac, te niz koraka koji je potrebno učiniti.

Ulazni podaci

U prvom redu nalazi se prirodni broj N ($5 \leq N \leq 500$).

U sljedećih N redaka nalaze se po dva broja R i S ($1 \leq R, S \leq N$) koji označavaju redak i stupac u kojem se pojedini tenk nalazi u trenutku kad je Mirkova mama zvala na ručak. Nijedna dva tenka se ne nalaze na istome polju.

Retci su numerirani odozgo prema dolje, a stupci slijeva na desno. Tenkovi su numerirani od 1 do N redom kojim su navedeni u ulazu.

Izlazni podaci

U prvi redak ispišite broj K – minimalni broj koraka.

U sljedećih K redaka ispišite broj B i znak C , odvojene razmakom. Broj B označava redni broj tenkića koji želite pomaknuti, a znak C označava smjer i to 'L' – lijevo, 'R' – desno, 'U' – gore, 'D' – dolje.

Napomena: Rješenje ne mora biti jedinstveno.

Bodovanje

Ako vaš program ispiše točan broj K , a niz koraka nije valjan dobiti ćete 60% bodova za taj test podatak.

4. TENKICI

Primjeri test podataka

ulaz

5
1 1
1 2
1 3
1 4
1 5

izlaz

10
1 D
2 D
3 D
4 D
1 D
2 D
3 D
1 D
2 D
1 D

ulaz

5
2 3
3 2
3 3
3 4
4 3

izlaz

8
1 R
1 R
2 U
2 U
4 D
4 D
5 L
5 L

ulaz

6
1 1
1 2
2 1
5 6
6 5
6 6

izlaz

8
2 R
2 D
3 D
3 R
4 U
4 L
5 L
5 U

5. BICIKLI

U jednoj dalekoj zemlji održava se biciklistička utrka. Zemlja ima N gradova označenih brojevima od 1 do N i M jednosmjernih cesta. Grad s brojem 1 označen je kao start, a grad s brojem 2 kao cilj.

Na koliko različitih načina možemo postaviti stazu za utrku? Dvije staze su različite ako ne idu istim cestama.

Ulazni podaci

U prvom redu nalaze se dva prirodna broja N i M ($1 \leq N \leq 10\,000$, $1 \leq M \leq 100\,000$) – broj gradova i broj cesta.

U sljedećih M redaka nalaze se po dva različita broja A i B . To znači da postoji jednosmjerna cesta koja vodi iz grada A u grad B . Moguće je da postoji više istosmjernih cesta koje povezuju iste parove gradova.

Izlazni podaci

U prvi i jedini red ispišite ukupan broj različitih staza. Ako taj broj ima više od 9 znamenaka ispišite samo zadnjih 9 znamenaka tog broja. Ako postoji beskonačno mnogo različitih staza, tada ispišite "inf".

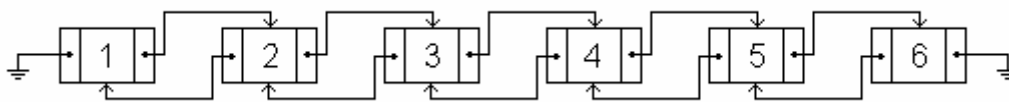
Primjeri test podataka

ulaz	ulaz	ulaz
6 7	6 8	31 60
1 3	1 3	1 3
1 4	1 4	1 3
3 2	3 2	3 4
4 2	4 2	3 4
5 6	5 6	4 5
6 5	6 5	4 5
3 4	3 4	5 6
	4 3	5 6
izlaz	izlaz	6 7
		6 7
3	inf	...
		...
		...
		28 29
		28 29
		29 30
		29 30
		30 31
		30 31
		31 2
		31 2
		izlaz
		073741824

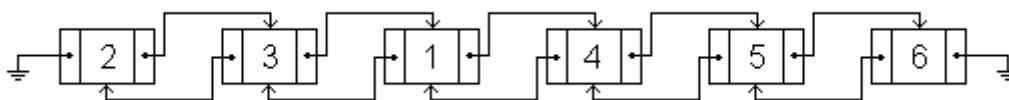
6. LISTA

Mirko je za rođendan, od tete iz Amerike, dobio novu dvostruko vezanu listu (primjer takve liste sa 6 čvorova se nalazi na donjoj slici). Lista se sastoji od N čvorova numeriranih od 1 do N . Lista podržava dvije operacije nad njom, a to su:

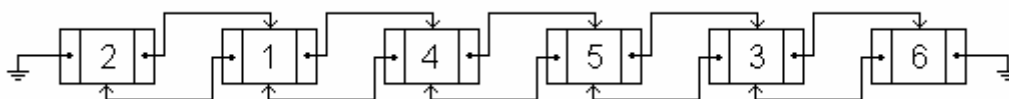
- A) Prebaci čvor s brojem X **ispred** čvora s brojem Y .
- B) Prebaci čvor s brojem X **iza** čvora s brojem Y .



Dvostruko vezana lista u početnom stanju.



Lista nakon operacije "A 1 4".



Lista nakon operacije "B 3 5".

Mirko se satima igrao s novom igračkom te je svaku operaciju zapisao na papir kako bi na kraju listu mogao vratiti u početno stanje (stanje u kojem su čvorovi poredani od 1 do N slijeva na desno).

Kada je došao taj trenutak, Mirko se zaprepastio shvativši da iz zapisanih podataka ne može obrnutim postupkom listu dovesti u početno stanje. Naime, Mirko ne zna gdje se čvor s brojem X nalazio prije svake operacije, već samo gdje se nalazi nakon operacije.

Kako je Mirko i dalje u šoku, vaš je zadatak da napišete program koji će iz Mirkovih zapisa odrediti minimalan niz operacija koji listu vraćaju u početno stanje.

Ulazni podaci

U prvom redu nalaze se dva prirodna broja N i K ($2 \leq N \leq 500\,000$, $0 \leq M \leq 100\,000$) – broj čvorova, te broj operacija koje je Mirko napravio.

U sljedećih M redaka nalazi se znak C , te dva broja X i Y . Znak C je 'A' ili 'B', te predstavlja o kojoj se operaciji radi, a X i Y su različiti brojevi čvorova nad kojima se operacija vrši.

Izlazni podaci

U prvi red ispišite broj K – minimalni broj operacija potreban za vraćanje liste u početno stanje.

U sljedećih K redaka ispišite tražene operacije u istom obliku kao što se nalaze u ulazu.

Napomena: Rješenje ne mora biti jedinstveno.

6. LISTA

Bodovanje

Ako vaš program ispiše točan broj K, a niz operacija nije valjan dobiti ćete 60% bodova za taj test podatak.

Primjeri test podataka

ulaz

2 1
A 2 1

izlaz

1
A 1 2

ulaz

4 3
B 1 2
A 4 3
B 1 4

izlaz

2
A 1 2
B 4 3

ulaz

6 5
A 1 4
B 2 5
B 4 2
B 6 3
A 3 5

izlaz

3
A 4 5
B 6 5
A 2 3