

 ispitni centar
**PRAVA
MJERA
ZNAŃJA**

**DRŽAVNO
TAKMIČENJE**

2019.

ŠIFRA UČENIKA

SREDNJA ŠKOLA

PROGRAMIRANJE

UKUPAN BROJ OSVOJENIH BODOVA



Test pregledala/pregledao

.....
.....
Podgorica, 20..... godine

Uputstva takmičarima

Ovo takmičenje sastoji se od rješavanja **3 problemska zadatka** u vremenu od **4 sata** (240 minuta). Zadatke je potrebno rješavati u jednom od sljedećih programskih jezika: Pascal, C, C++ ili Java. Takmičari koji koriste Pascal moraju programirati u programskom alatu **FreePascal**. Takmičari u C-u i C++-u moraju koristiti programski alat **CodeBlocks**. Za programski jezik Java predviđena je upotreba platforme **Eclipse**. Dozvoljeno je koristiti editor po izboru i pomoću navedenih alata prevoditi izvorni kod u izvršnu datoteku.

Tokom takmičenja **ne smijete komunicirati** ni sa jednom osobom, osim dežurne osobe takmičenja. To znači da morate **raditi samostalno i ne smijete koristiti Internet**. Takođe, zabranjena je upotreba bilo kakvih ranije napisanih programa ili dijelova programa.

Po isteku vremena predviđenog za takmičenje, na desktopu u folderu sa imenom **Takmicenje2019** moraju se nalaziti datoteke sa snimljenim izvornim kôdovima rješenja. Nakon takmičenja, komisija će testirati vaša rješenja na ranije izabranim test podacima i dodijeliti vam određeni broj bodova. Na kraju svakog zadatka dati su primjeri test podataka. Ti primjeri služe da bi vam tekst zadatka 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 na službenim podacima za testiranje.

Zadaci ne nose 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. Ukupan broj bodova na nekom zadatku jednak je zbiru bodova test podataka koji se poklapaju sa službenim rješenjem. Ukupan broj bodova jednak je zbiru bodova na svim zadacima.

Sve informacije o zadacima (ime zadatka, vremensko i memorijsko ograničenje, način bodovanja) možete naći na uvodnoj stranici s naslovom *Zadaci*. Ako vam nije jasno nešto u vezi načina organizacije ovog takmičenja, odmah postavite pitanje dežurnom da vam to razjasni.

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

VAŽNO za C/C++!

Glavni program (glavna funkcija) **mora** biti deklarisan 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()` deklariranu u `<stdio.h>` i `<cstdio>`, koja ima i veće mogućnosti primjene,

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()`.

VAŽNO za jezik Java!

Ne kreirajte pakete za vaše zadatke, već koristite podrazumijevani (default) paket.

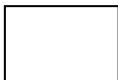
Nepoštovanje ovih pravila ili nepridržavanje formata izlaznih podataka rezultiraće nepovratnim gubitkom bodova. Nemojte štampati ništa što se u zadatku ne traži, kao npr. poruke tipa 'Rjesenje je:' ili 'Unesite brojeve' i slično!

Srećno i uspješno takmičenje!

Zadaci

Zadatak	Zadatak1	Zadatak2	Zadatak3
Izvorni kôd	zadatak1.java zadatak1.pas zadatak1.c zadatak1.cpp	zadatak2.java zadatak2.pas zadatak2.c zadatak2.cpp	zadatak3.java zadatak3.pas zadatak3.c zadatak3.cpp
Memorijsko ograničenje	256 MB	256 MB	256 MB
Vremensko ograničenje (po test podatku)	1 sekunda	2 sekunde	2 sekunde
Ukupno bodova	30	35	35

Zadatak 1



Ana želi da pozitivan cio broj predstavi u obliku proizvoda jednog ili više pozitivnih brojeva. Ona želi da razlika između bilo koja dva činioca u proizvodu bude najviše 1. Ana smatra da su dva predstavljanja ista ako imaju isti broj činilaca i jedno od njih se može dobiti permutovanjem elemenata drugog predstavljanja.

Ulazni podaci

Jedini red ulaza sadrži jedan cio broj n ($1 \leq n \leq 10^{18}$).

Izlazni podaci

U prvom redu izlaza štampati broj predstavljanja broja n ili broj -1 (minus 1) ako je odgovor beskonačno. Ako je broj predstavljanja konačan, štampati svako predstavljanje. U svakom redu prvo štampati broj k_i činilaca u predstavljanju a zatim i k_i brojeva – činioce predstavljanja u bilo kom poretku. Ne zaboravite da se predstavljanja koja se razlikuju samo u redosljedu činilaca smatraju jednakim.

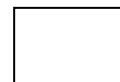
Test primjeri

Ulaz	Izlaz
12	3 1 12 3 2 3 2 2 4 3
1	-1

Rješenje:

Matematička formulacija zadatka je sljedeća: $n = \prod x_i$, uz uslov da je $\max|x_i - x_j| \leq 1$. To znači da se broj n može zapisati u obliku $n = x^a(x+1)^b$. Otuda je $x = \lfloor \sqrt[a+b]{n} \rfloor$. Moguće vrijednosti za $a+b$ su od 1 do $\log_2 n$, pa je dovoljno proći kroz sve mogućnosti za $a+b$ i provjeriti da li važi $n = x^a(x+1)^b$. Brojevi x i $x+1$ su uzajamno prosti, pa je $a = \max\{i | n \text{ je djeljiv sa } x^i\}$ i $b = \max\{i | n \text{ je djeljiv sa } (x+1)^i\}$.

Zadatak 2



Ratko i Ana pišu igru za mobilne telefone u kojoj se superheroj ALFA bori protiv n zlih dronova. ALFA je na početku igre zauzeo poziciju u koordinatnom početku, dok se i -ti dron nalazi na poziciji (x_i, y_i) , x_i i y_i su cijeli brojevi. Za jednu sekundu ALFA može specijalnom puškom uništiti tačno jednog drona, dok se svi preostali dronovi mogu pomjeriti na jednu od susjednih 8 pozicija po horizontali, vertikali ili dijagonali, pri čemu se više dronova može naći na istoj poziciji. Cilj dronova je da zarobe ALFU, dok ALFA ima suprotan zadatak: uništite sve dronove prije nego što ga zarobe tj, prije nego što dođu do pozicije $(0, 0)$. ALFA nikada ne promašuje i svakim hicem obara jednog drona.

Napišite program za Ratka i Anu koji pokazuje u kom poretku treba uništavati dronove da ne bi stigli do ALFE, ili koji nam govori da ALFI nema spasa i da će biti zarobljen.

Ulazni podaci

Prvi red ulaza sadrži broj n – broj dronova u igri ($1 \leq n \leq 10^5$). U sljedećih n redova zadaju se pozicije dronova – po dva cijela broja x_i i y_i ($|x_i|, |y_i| \leq 10^5$). U tački $(0, 0)$ nema dronova.

Izlazni podaci

U jedinom redu izlaza štampati n brojeva – brojeve dronova u poretku kojim ih ALFA mora uništavati. Ako se neki dron probije do tačke $(0, 0)$, u jedini red izlaza štampati broj „-1“ (bez navodnika). Ako postoji više rješenja, štampati bilo koje od njih.

Test primjeri

Ulaz	Izlaz
3 0 1 -2 3 2 2	1 3 2
3 0 1 -2 2 2 2	-1

Rješenje:

Ako se dron optimalno kreće ka ALFI, potrebno mu je $\max(|x_i|, |y_i|)$ koraka do pozicije $(0, 0)$. Neka je to rastojanje drona do cilja. ALFI je najbolje da prvo gađa najbliže dronove. Sortirajmo dronove po rastojanjima od cilja u rastućem poretku. ALFA ne može uništiti drona x ako je rastojanje do drona x manje od broja dronova čije su rastojanja ne veća od rastojanja x . Prođimo kroz dronove u sortiranom nizu i

provjerimo da li je $\max(|x_i|, |y_i|) \geq i$. Ako je to tačno za sve dronove, štampamo dronove u poretku uvećanja rastojanja. U suprotnom, rješenje ne postoji i štampamo -1.

Zadatak 3



Ana i Ratko razvijaju još jednu igru za mobilne telefone. U njoj naš superheroj ALFA mora kroz lavirint da prenese deblo neke dužine. Lavirint je predstavljen kao tabela od $m \times n$ polja. Svako polje je ili prazno ili sadrži zid lavirinta. Debla koje nosi ALFA mogu biti dužine k , ali moraju zauzimati k uzastopnih polja lavirinta u jednoj liniji. Deblo ne može zauzeti polje u kojem se nalazi zid. ALFA je veoma jak, pa može prenijeti deblo proizvoljne dužine. U lavirint se može ući u bilo kom polju sa lijeve ivice, ali deblo mora biti paralelno toj ivici. Da bi izašao iz lavirinta, mora se naći u bilo kom polju sa desne ivice lavirinta, pri čemu deblo mora da bude paralelno toj ivici. Ako ALFA drži deblo paralelno jednoj ivici lavirinta, može ga pomjeriti jedno polje duž linije debbla ili ga može uhvatiti za jedan kraj, podići ga u vazduh i položiti ga paralelno nekoj od ivica lavirinta. Naravno, ne smije spustiti deblo u polje koje sadrži zid. Vaš zadatak je da odredite najveću moguću dužinu debbla koje ALFA može prenijeti kroz lavirint.

Ulazni podaci

Prvi red ulaza sadrži brojeve n i m – broj redova i kolona lavirinta ($1 \leq n, m \leq 300$). Sjedećih n redova sadrže po m simbola; u i -tom redu simbol na poziciji j je '#', ako je to polje zid ili '.' ako je to polje slobodno.

Izlazni podaci

U jedinom redu izlaza štampati jedan cio broj – najveću dužinu debbla koje ALFA može prenijeti kroz lavirint. Ako to nije moguće, štampati broj 0.

Test primjeri

Ulaz	Izlaz
3 5 ...## .#.#. ##...	2

Rješenje:

Ako je moguće prenijeti deblo dužine k , tada je to moguće uraditi i za debbla dužina manjih od k . Traženo k je moguće odrediti binarnim traženjem. Provjeru da li je moguće prenijeti deblo dužine k možemo uraditi primjenom DFS-a na grafu čija su stanja opisana pozicijom gornjeg lijevog polja debbla i smjerom (istok-zapad ili sjever-jug). Iz svakog stanja imamo 6 prelaza: dva u smjeru debbla i 4 rotacije za po 90 stepeni oko jednog od krajeva debbla. Za provjeru da ne ulazimo u unutrašnji zid

lavirinta mogu se naći prefiksne sume po svakom redu i koloni. Ovo rješenje ima složenost $O(n \cdot m \cdot \log(\min(n, m)))$.