



ispitni centar

PRAVA
MJERA
ZNANJA

DRŽAVNO
TAKMIČENJE

2020.

SREDNJA ŠKOLA

PROGRAMIRANJE

Autorka/autor testa

Recenzentkinja/recenzent

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 ili TurboPascal**. Takmičari u C-u i C++-u moraju koristiti programske alate **CodeBlocks, DJGPP, DevCpp ili GCC**. 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 **Takmicenje2020** 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()`.

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!

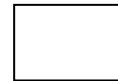
Srećno i uspješno takmičenje!

Zadaci

Zadatak	Zadatak1	Zadatak2	Zadatak3
Izvorni kôd	Zadatak1.java zadatak1.pas zadatak1.c zadatak1.cpp	Zadatak2.java zadatak2.pas zadatak2.c zadatak2.cpp	Yadatak3.java zadatak3.pas zadatak3.c zadatak3.cpp
Memorijsko ograničenje	64 MB	256 MB	64 MB
Vremensko ograničenje (po test podatku)	1 sekunda	1 sekunda	1 sekunde
Broj test podataka	10	10	10
Broj bodova (po test podatku)	3	3.5	3.5
Ukupno bodova	30	35	35

Napomena: Program u C-u i C++-u treba kompajlirati sa sljedećim opcijama: `-O2 -lm -static`, a program u Pascalu sa `-O1 -XS`.

Zadatak 1 – Kvadrati



Helena je u školi počela da izučava Dekartov pravougli koordinatni sistem, pa je vježbajući nacrtala prvi kvadrant tog sistema i u njemu kreirala mrežu kvadratića stranice 1 cm. Helenina mlađa sestra Anja je na tom papiru nacrtala više kvadrata različitih stranica čija se tjemena nalaze u tjemenu kvadratića. Helena se prvo naljutila na Anju, ali je posmatrajući Anjin crtež brzo shvatila da može napraviti zanimljivu igru za svoje drugare iz odjeljenja. Naime, ona je zamislila jedan broj S i sada je zanima koliko ima kvadrata koji zadovoljavaju sljedeće uslove:

- površina svakog kvadrata nije veća od S ;
- koordinate svih tjemena kvadrata u Dekartovom koordinatnom sistemu su cijeli brojevi;
- svi kvadrati imaju različite površine.

Pomozite Heleni da za dato S odredi koliko najviše postoji takvih kvadrata.

Ulazni podaci

Prvi red ulaza sadrži cio broj S , $0 < S < 20\,000\,000$.

Izlazni podaci

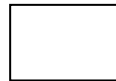
Štampati jedan cio broj – maksimalan broj traženih kvadrata

Test primjeri

Ulaz	Izlaz	Objašnjenje
4	3	Na primjer, traženi kvadrati su: K1 sa tjemenuima $(0,0)$, $(1,0)$, $(1,1)$ i $(0,1)$; K2 sa tjemenuima $(3,0)$, $(4,1)$, $(3,2)$ i $(2,1)$; K3 sa tjemenuima $(5,0)$, $(7,0)$, $(7,2)$ i $(5,2)$. Ovo nisu jedini takvi kvadrati.

Rješenje: Treba prebrojati kvadrate čija dužina stranice c zadovoljava nejednakost $c^2 \leq S$. Kako stranica kvadrta ima tjemena u cjelobrojnim tačkama, to treba da odredimo parove nenegativnih cijelih brojeva a i b takvih da je $a^2 + b^2 \leq S$, jer je po Pitagorinoj teoremi c dužina hipotenuze dok su a i b dužine kateta pravouglog trougla (obratite pažnju da trougao može biti "degenerisan"). Pretragu možemo obaviti dvostrukom petljom tako da važi $0 \leq a^2 \leq S$, $1 \leq b \leq a$ i $a^2 + b^2 \leq S$. Za par (a, b) možemo pamtiiti vrijednosti u nizu $x[a^2 + b^2]$, pa je dovoljno da na kraju prebrojimo koliko ima nenultih elemenata u nizu x .

Zadatak 2 – Duži



U Dekartovom koordinatnom sistemu date su dvije različite horizontalne prave. Na prvoj pravoj dato je n tačaka sa cjelobrojnim koordinatama x : $a_1, a_2, \dots, a_n$. Na drugoj pravoj takođe je dato n tačaka sa cjelobrojnim koordinatama x : $b_1, b_2, \dots, b_n$. Marija je nacrtala skup od n duži, tako što je i -tu tačku sa prve prave spojila sa i -tom tačkom sa druge prave, $i=1, \dots, n$. Sada je zanimaju podskupovi datog skupa duži tako da se nijedan par duži u podskupu ne siječe. Pomozite Mariji i odredite koliko najviše elemenata ima takav podskup.

Ulazni podaci

Prvi red ulaza sadrži cio broj n , $1 < n < 10^5$. Drugi red ulaza sadrži brojeve $a_1, a_2, \dots, a_n$ razdvojene sa po jednim blankom, među kojima nema jednakih, $1 < a_i < 10^6$. Treći red ulaza sadrži brojeve $b_1, b_2, \dots, b_n$ razdvojene sa po jednim blankom, među kojima nema jednakih, $1 < b_i < 10^6$.

Izlazni podaci

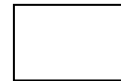
Jedini red izlaza sadrži cio broj – maksimalan broj duži u podskupu.

Test primjeri

Ulaz	Izlaz	Objašnjenje
4 12 8 2 5 4 2 3 6	2	Date su ukupno 4 duži: $(12, 4)$, $(8, 2)$, $(2, 3)$ i $(5, 6)$. Na primjer, podskup $\{(12, 6), (8, 2)\}$ ima 2 elementa, a svaki podskup koji ima više od 2 elementa sadrži duži koje se sijeku.

Rješenje: Posmatrajmo niz parova $(a_1, b_1), (a_2, b_2), \dots, (a_n, b_n)$. Sortirajmo niz po prvoj koordinati para. Posmatrajmo niz d drugih koordinata dobijenih iz sortiranog niza parova. Rješenje zadatka je broj elemenata u najdužem rastućem podnizu niza d . Najduži rastući podniz je dobro poznat zadatak. Može se riješiti za $O(n^2)$ i to rješenje nosi 30%. Za 100% potrebno je koristiti binarno traženje, sa složnošću $O(n \log n)$.

Zadatak 3 – Bakterije



Biolozi Marko i Ana uzgajaju novi eksperimentalnu bakterijsku kulturu za poljoprivedni kombinat. Za izvođenje eksperimenata koriste tablu oblika pravougaonika dimenzije $W \times H$ centimetara koju predstavljamo matricom koja sadrži $W \times H$ polja. Za svako polje naši biolozi znaju da li je pogodno za širenje kulture ili ne. Eksperiment se izvodi na sljedeći način. Prvo biraju jednu dijagonalu i sva polja koja nisu pogodna na toj dijagonali, ako ih ima, dodatno obrađuju tako da postanu pogodna. U svako polje odabrane dijagonale postavljaju bakterije, koje počinju da se šire na susjedna pogodna polja, gdje se susjedima smatraju polja gore, dolje, lijevo i desno. Bakterije se šire sve dok je to moguće. Pomozite Marku i Ani da odrede koju dijagonalu treba da izaberu za svoj eksperiment tako da na kraju dobiju najveći mogući broj polja zahvaćenih bakterijom.

Ulazni podaci

U prvom redu ulaza nalaze se prirodni brojevi W i H ($3 \leq W, H \leq 2000$) – dimenzije table. Sljedećih W redova sadrže po H simbola bez razmaka koji opisuju Aninu i Markovu tablu. Simbol 'a' predstavlja polje pogodno za bakterije, a simbol 'b' polje koje nije pogodno.

Izlazni podaci

Štampati jedan cio broj – najveći mogući broj polja zahvaćenih bakterijom.

Test primjeri

Ulaz	Izlaz	
5 6 bababb abbabb bbbaab abbaaa babbaa	15	Simbol x označava bakterije. Boldovana i podvučena je izabrana dijagonala. bxxbb xxbb xbxb bbxb bbbbb babb
8 8 bbbabbba bbbabbba bbbabbbb aaaaaaaa bbbabbbb bbbabbab abbabaaa aababbab	26	 bbxbba bxbba bbxbbbb xxx bbxbbb bbxbxb abbxbxx aabxbbx

Napomena: U 50 % primjera važiće uslov $W, H \leq 300$.

Rješenje: U primjerima za testiranje za 50% bodova važi $W, H \leq 300$. Algoritam je sledeći: za svaku dijagonalu označimo polja na dijagonali i njima susjedna polja kao pogodna i iz nekog polja dijagonale pokrenemo DFS ili BFS koji broji do koliko je pogodnih polja moguće doći iz početnog. Ako je dobijena vrijednost veća od do sada najbolje vrijednosti, onda nju pamtimo. Ostale testove moguće je riješiti tako da pretprocesiramo podatke koji će za svaku dijagonalu smanjiti broj operacija. Prije izbora dijagonala možemo pronaći komponente pogodnih polja i za svaku od njih pamtimo njenu površinu. Takođe, za svako pogodno polje pamtimo redni broj komponente kojoj pripada. Komponente možemo odrediti ponovo primjenom DFS ili BFS obilaska. Sada za svaku dijagonalu možemo odrediti broj polja koje bakterije zauzmu tako što prolazimo po svim poljima na dijagonali i njima susjednim poljima. Ako je polje nepogodno, ono postaje pogodno i povećavamo broj polja za 1. Ako je polje pogodno i pripada komponenti koju još nismo uračunali, na broj polja biljaka dodajemo veličinu odgovarajuće komponente i pamtimo da smo tu komponentu uračunali.