

Županijsko natjecanje iz informatike

Srednja škola

Prva podskupina (1. i 2. razred)

15. veljače 2019.

ime zadatka	STRANICE	PARCELA	SPUST
vremensko ograničenje	1 sekunda	1 sekunda	2 sekunde
memorijsko ograničenje	512 MiB	512 MiB	512 MiB
broj bodova	40	50	60
	150		



Ministarstvo
znanosti i
obrazovanja



Agencija za odgoj i obrazovanje



**HRVATSKI SAVEZ
INFORMATIČARA**

Upute za natjecatelje

Sastavni dio kompleta zadataka su i ove upute te uvodna stranica na kojoj se nalaze važni podatci o zadacima. Molimo vas da i jedno i drugo pažljivo pročitate. Na ostalim stranicama nalaze se tri zadatka. Prilikom rješavanja zadataka preporučuje se korištenje olovke i papira za skiciranje i razradu algoritma.

Nakon završetka natjecanja, članovi županijskih povjerenstava poslat će izvorne kodove vaših rješenja na automatsku evaluaciju. Nakon završetka evaluacije za vašu županiju, rezultati evaluacije bit će dostupni putem sustava za evaluaciju na adresi <https://srednje.hsin.hr>.

Žalbe na rezultate evaluacije možete uputiti direktno Županijskom povjerenstvu (koje će vas obavijestiti o roku i načinu predaje žalbi) ili Državnom povjerenstvu na adresu dp-informatika@azoo.hr. Rok za predaju žalbi Državnom povjerenstvu istječe *u ponoć na dan natjecanja*. Rok žalbi Državno povjerenstvo može dodatno produžiti u slučaju većih problema sa zadacima ili sustavom za evaluaciju.

Ulazni i izlazni podatci

Kod svakog pojedinog zadatka obratite pozornost na sekcije *Ulazni podatci* i *Izlazni podatci*. Tu su definirana pravila vezana uz format ulaznih i izlaznih podataka koji mora biti strogo poštovan kako bi vaša rješenja bila ispravno evaluirana. Vaš program sa standardnog ulaza mora očekivati samo zadane ulazne podatke, a na standardni izlaz ispisivati samo tražene izlazne podatke bez ikakvih dodatnih poruka. Ako vaš program bude čekao na unos nečeg drugog osim ulaznih podataka ili ispisivao nešto drugo osim izlaznih podataka (npr. *Unesite brojeve...*, *Rješenje je...* i slično), nećete dobiti bodove jer evaluator to ne očekuje.

Vaši programi ne smiju pristupati datotekama ili ih kreirati te ne smiju pokretati nove procese ili koristiti više od jedne dretve (threada).

Prilikom rješavanja nekog zadatka i testiranja njegovog rješenja preporučuje se korištenje operatora redirekcije ulaza kako ne biste više puta nepotrebno unosili podatke preko tipkovnice. Na primjer, ulazne podatke za neki od oglednih primjera iz teksta zadatka možete spremirati u tekstualnu datoteku i testirati vaš program tako da ga pokrećete iz komandne linije na sljedeći način (pretpostavimo da se zadatak zove „Neboder“):

```
neboder < primjer.txt
```

Znak < je operator redirekcije ulaza i sve što se nalazi u datoteci `primjer.txt` bit će proslijeđeno vašem programu kao da je uneseno preko tipkovnice.

U slučaju Pythona potrebno je eksplicitno pozvati odgovarajući prevoditelj. Na primjer:

```
C:\Python27\python neboder.py < primjer.txt
```

Bodovanje

Da bi program koji rješava neki zadatak dobio maksimalan broj bodova, primijenjeni algoritam mora biti najprije točan, ali i efikasan tj. brz. Test podatci unaprijed su osmišljeni tako da će programi koji koriste manje efikasne, ali valjane algoritme, također dobiti određeni broj bodova (npr. od ukupno 60 bodova, vrlo spor algoritam dobit će npr. 20 bodova, dok će dobar algoritam, ali ne i najbolji, dobiti npr. 40 bodova). Jako brz program koji ne daje točne rezultate, naravno, ne donosi bodove. Valjanost algoritma je na prvom mjestu, a brzina izvršavanja na drugom.

Različite zadatke možete rješavati u različitim jezicima, imajući na umu da će rješenja u Pythonu, zbog prirode jezika, biti sporija od ekvivalentnih rješenja u drugim dozvoljenim jezicima i da bi zato mogla na nekim test podacima premašiti vremensko ograničenje.

U (nekim) zadacima obratite pažnju na sekciju *Bodovanje*. Ona opisuje dio test podataka (te pripadni broj bodova) koji su potencijalno lakše rješivi jer sadrže manje brojeve ili su na neki način specifični.

Upute za natjecatelje

Evaluacija

Bodove za pojedini test podatak dobit će samo oni programi koji budu generirali točan rezultat unutar navedenog vremenskog i memorijskog ograničenja, te regularno završe svoje izvođenje. Točnije, program se treba izvršiti do kraja. U programskim jezicima C/C++ to znači do `return 0;` na kraju funkcije `main` koja treba biti deklarirana kao `int main()`, ili do naredbe `exit(0);`.

Ne trebate predati izvršnu (`exe`) datoteku već samo predajete datoteku s izvornim kodom. Imena datoteka moraju odgovarati imenima zadataka, a ekstenzija datoteke mora odgovarati programskom jeziku i standardu u skladu s donjom tablicom (`.c`, `.cpp`, `.cxx` ili `.py`). Npr. ako se zadatak zove „Neboder“ i ako koristite C++11, predat ćete datoteku `neboder.cxx`. Na temelju odgovarajuće ekstenzije, za jezike C/C++ sustav za evaluaciju iz vašeg će izvornog kôda kreirati izvršnu datoteku na sljedeći način:

```
C          gcc -DEVAL -O2 -o neboder neboder.c -lm
C++       g++ -DEVAL -O2 -o neboder neboder.cpp
C++11    g++ -DEVAL -std=c++11 -O2 -o neboder neboder.cxx
```

Za rješenja u Pythonu, kako bi evaluator prepoznao koristite li verziju 2 ili verziju 3, prva linija u kodu mora točno odgovarati jednom od sljedećeg:

- `#!/usr/bin/python2`
- `#!/usr/bin/python3`

Za evaluaciju će se koristiti verzije prevoditelja 2.7 odnosno 3.5, a prije izvođenja vaš kod bit će preveden u bytecode naredbom `python -m py_compile neboder.py`.

Računalo na kojem se vrši evaluacija je Linux računalo s Ubuntu 16.04 LTS 64-bitnim operativnim sustavom i sljedećim verzijama prevoditelja: `gcc 4.9`, `g++ 4.9`. Preporučujemo da svoja rješenja obavezno isprobate sa `gcc` ili `g++` prevoditeljima s gore navedenim opcijama, osobito ako koristite neki drugi alat (npr. Microsoft Visual Studio) tijekom natjecanja. Da bi vaše rješenje bilo uspješno prevedeno, morate koristiti samo standardne biblioteke, tj. ne smijete koristiti naredbe i funkcije specifične za Windows.

Primjeri pravilno napisanih programa

Zadatak: Napišite program koji će zbrojiti i oduzeti dva cijela broja.

Ulaž: U prvom retku nalaze se dva cijela broja A i B, međusobno odvojena jednim razmakom.

Izlaž: U prvi redak ispišite zbroj, a u drugi redak razliku brojeva A i B.

C	C++	Python 2
<pre>#include <stdio.h> int main(void) { int a, b; scanf("%d%d", &a, &b); printf("%d\n", a + b); printf("%d\n", a - b); return 0; }</pre>	<pre>#include <iostream> using namespace std; int main(void) { int a, b; cin >> a >> b; cout << a + b << endl; cout << a - b << endl; return 0; }</pre>	<pre>#!/usr/bin/python2 a, b = map(int, raw_input().split()) print a + b print a - b</pre>
		<p>Python 3</p> <pre>#!/usr/bin/python3 a, b = map(int, input().split()) print(a + b) print(a - b)</pre>

Upute za natjecatelje

Česte pogreške

Donosimo listu **čestih i nepotrebnih pogrešaka** na natjecanjima ovog tipa. Sve od sljedećeg može rezultirati time da će evaluator dodijeliti nula bodova vašem rješenju.

- **Manjak inicijalizacije lokalnih varijabli:** U jezicima C i C++ lokalne varijable (što uključuje i varijable deklarirane unutar funkcije `main`) ne inicijaliziraju se automatski te njihova početna vrijednost nije definirana. Obavezno inicijalizirajte vrijednosti svih lokalnih varijabli, inače je moguće da vaš program dobro radi lokalno, a dobije nula bodova na sustavu za evaluaciju.
- **Rješenje alocira premala polja:** Ako ne alocirate dovoljno velika polja, moguće je da vaš program dobro radi za primjere iz teksta zadatka, a dobije nula bodova na evaluatoru. Na primjer, ako s ulaza trebate pročitati niz od najviše 1000 znakova, u jeziku C potrebno je alocirati polje od najmanje 1001 elementa (`char s[1001]`).
- **Rješenje alocira prevelika polja:** Nepotrebno glomazna polja mogu prouzročiti da vaše rješenje prekorači memorijsko ograničenje i dobije nula bodova na evaluatoru. Primjerice, u jeziku C polje deklarirano sa `int x[1024][1024][100]` koristi 400MiB memorije.
- Manjak direktive `from sys import exit` u slučaju korištenja naredbe **`exit`** u jeziku Python.
- Korištenje **`conio.h`** u jezicima C i C++. Korištenje sintakse, tipova i funkcija specifičnih za Microsoftove C i C++ prevoditelje, npr. tip podataka **`int64`**. Korištenje naredbi **`itoa`** (koje nema pod Linuxom) ili **`atoi`** (koja radi malo drugačije pod Linuxom). Umjesto njih preporučamo korištenje funkcija `scanf` i `sprintf`.
- Rješenja čekaju na pritisnutu tipku nakon ispisa rezultata, npr. zbog naredbe **`system("pause")`**.
- Rješenja ispisuju rezultate u **pogrešnom formatu**, npr. ispisuju dva broja svaki u svoj redak umjesto u istom retku.
- Rješenja ispisuju **višak podataka** na standardni izlaz, npr. **debug informacije** ili poruke poput "Upišite broj:", "Rješenje je:" i slično.
- Glavna funkcija u jezicima C i C++ definirana je kao `void main()` ili nema naredbe `return 0;`.
- Manjak potrebnih `include` direktiva u jezicima C i C++.

Dodatno, natjecateljima koji koriste C i C++ savjetujemo da prilikom testiranja obavezno uključite optimizacijsku opciju koja se koristi na sustavu na evaluaciju (`-O2`) te opcije koji upozoravaju na moguće probleme u kodu (primjerice `-Wall -Wextra`). Niže je ilustracija takvog prevođenja jednog rješenja sa Županijskog natjecanja 2016. godine. Prvo upozorenje koje je prijavio prevoditelj je lažna uzbuna, dok drugo ukazuje na stvarnu grešku (neinicijalizirana varijabla) zbog koje je natjecatelj nepotrebno izgubio bodove.

```
$ g++ -O2 -o poli.exe poli.cpp -Wall -Wextra
poli.cpp: In function 'int main()':
poli.cpp:16:20: warning: comparison between signed and unsigned integer expressions [-Wsign-compare]
   for (int i = 0; i < in.length(); ++i)
                       ^
poli.cpp:36:29: warning: 'curr' may be used uninitialized in this function [-Wmaybe-uninitialized]
   num[(curr == 0 ? 1 : curr) - 1] = val;
```

Niz prirodnih brojeva 1, 2, 3, 4, ... zapisali smo u trokut tako da smo u prvi red trokuta upisali jedan broj, u drugi red dva broja, u treći red tri broja, i tako dalje:

```

      1
     2 3
    4 5 6
   7 8 9 ...

```

Na ovaj način ispunili smo točno N redova trokuta. Potom smo odlučili više puta obrisati neku stranicu trokuta, tj. niz brojeva na jednom od triju rubova trokuta. Stranice trokuta označavamo slovima A, B, C u smjeru kazaljke na satu: A označava lijevu, B desnu, a C donju stranicu trokuta. Na sljedećim slikama prikazano je brisanje stranica A, B, C redom (podebljani brojevi se brišu):

<u>1</u>			
<u>2</u> 3	<u>3</u>		
<u>4</u> 5 6	5 <u>6</u>	5	5
<u>7</u> 8 9 10	8 9 <u>10</u>	<u>8</u> <u>9</u>	
<i>brišemo stranicu A</i>	<i>brišemo stranicu B</i>	<i>brišemo stranicu C</i>	<i>ostaje samo broj 5</i>

Uočite da se nakon svakog brisanja dimenzija stranice trokuta smanjuje za jedan. Brisati smo mogli i drugačije: primjerice, od istog početnog trokuta nakon brisanja stranica C, B, C ostao bi samo broj 2.

Napišite program koji na temelju oznaka stranica koje se brišu računa **zbroj preostalih** brojeva u trokutu **nakon svakog brisanja**.

ULAZNI PODATCI

U prvom retku nalazi se prirodan broj N ($2 \leq N \leq 4000$), broj redova trokuta.

U drugom retku nalazi se niz od $N - 1$ znakova A, B ili C bez razmaka: stranice trokuta redom kojim se brišu.

IZLAZNI PODATCI

Za svako brisanje u zaseban redak ispišite traženi zbroj preostalih brojeva.

PRIMJERI TEST PODATAKA

ulaz

4
ABC

izlaz

41
22
5

ulaz

4
CBC

izlaz

21
11
2

ulaz

5
BABA

izlaz

85
61
33
13

Milan je zaposlio Mirka da u njegovom gradu izgradi tri nove zgrade. Svaka od zgrada koje Mirko mora izgraditi imat će pravokutan tlocrt dimenzija koje mu Milan zada.

Da bi Mirko započeo s gradnjom, mora najprije kupiti odgovarajuću parcelu. S obzirom da želi na kupnju parcele potrošiti što manje novca, tražit će parcelu s najmanjom površinom na kojoj može izgraditi sve tri zgrade. Unutar parcele zgrade može smjestiti bilo kako okrenute i raspoređene. Tlocrti zgrada smiju se dodirivati, ali ne i preklapati.

Pomozite Mirku tako što ćete izračunati površinu najmanje parcele na kojoj može izgraditi tri zgrade ako vam kaže dimenzije zgrada koje mu je Milan zadao.

ULAZNI PODATCI

U prvom retku nalaze se dva prirodna broja a_1, b_1 , ($1 \leq a_1, b_1 \leq 10\,000$), dimenzije tlocrta prve zgrade.

U drugom retku nalaze se dva prirodna broja a_2, b_2 ($1 \leq a_2, b_2 \leq 10\,000$), dimenzije tlocrta druge zgrade.

U trećem retku nalaze se dva prirodna broja a_3, b_3 ($1 \leq a_3, b_3 \leq 10\,000$), dimenzije tlocrta treće zgrade.

IZLAZNI PODATCI

U jedini redak ispišite traženi broj iz teksta zadatka.

BODOVANJE

U test primjerima vrijednim 30% bodova, sve dimenzije bit će manje ili jednake 5.

U test primjerima vrijednim idućih 20% bodova, sve dimenzije bit će manje ili jednake 30.

U test primjerima vrijednim idućih 20% bodova, sve dimenzije bit će manje ili jednake 200.

U test primjerima vrijednim posljednjih 30% bodova nema dodatnih ograničenja.

Također će u 30% ukupnog broja primjera vrijediti da su **tlocrti svih zgrada kvadrati** kao u drugom oglednom primjeru niže.

PRIMJERI TEST PODATAKA

ulaz

1 5
3 3
4 2

izlaz

24

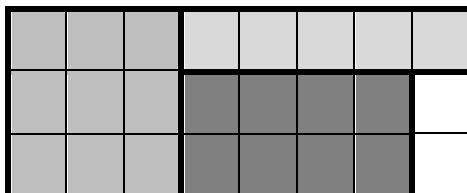
ulaz

3 3
2 2
1 1

izlaz

15

Pojašnjenje prvog primjera: Mirko će kupiti parcelu dimenzija 3×8 na kojoj može izgraditi zadane zgrade kao na slici. Na parceli manje površine ne može izgraditi tražene zgrade.



Mirko je ove godine otišao na skijanje na planinu koju možemo prikazati kao matricu dimenzija $N \times M$. Broj u svakom polju matrice označava visinu u metrima odgovarajućeg kvadratnog metra planine.

Mirko je odlučio svaki dan spustiti se jednom niz planinu. Njegov spust opisujemo nizom polja matrice kroz koja se spušta. Za svako polje Mirkovog spusta (osim početnog) mora vrijediti da je ono strogo manje visine od prethodnog polja spusta, te da s prethodnim poljem dijeli stranicu. Duljinu spusta definiramo kao broj polja matrice koja spust sadrži.

Trenutačno je zima te je na cijeloj planini snijeg, no dolazi proljeće i snijeg se polako počinje topiti. Mirko zna da se na manjim nadmorskim visinama snijeg brže topi. Preciznije, nakon i -tog dana njegovog skijanja više neće biti snijega na dijelovima planine visine i metara.

Mirko želi što bolje iskoristiti ovogodišnje skijanje pa će svakog dana odskijati najdulji mogući spust. Pomozite Mirku tako da za zadanu planinu ispišete duljinu najduljeg spusta koji Mirko može odskijati za svaki od prvih K dana skijanja, ako znate da Mirko nikad neće pokušavati skijati na dijelovima planine na kojima nema snijega.

ULAZNI PODATCI

U prvom retku nalaze se prirodni brojevi N , M i K ($1 \leq N$, $M \leq 1000$, $1 \leq K \leq 1\,000\,000$) iz teksta zadatka.

U svakom od idućih N redaka nalazi se M brojeva a_{ij} ($1 \leq a_{ij} \leq 1\,000\,000$) koji opisuju visine dijelova planine.

IZLAZNI PODATCI

Potrebno je ispisati K redaka. U i -ti redak ispišite duljinu najduljeg spusta kojeg Mirko može skijati na i -ti dan.

PRIMJERI TEST PODATAKA

ulaz

```
4 4 10
1 3 4 5
7 8 1 9
2 3 4 2
5 1 2 3
```

izlaz

```
5
4
4
3
2
2
2
1
1
0
```

ulaz

```
3 5 3
1 2 1 3 2
3 5 4 3 2
3 4 2 1 2
```

izlaz

```
4
4
3
```

[Pojašnjenje se nalazi na idućoj stranici.]

Zadatak SPUST

2 sekunde / 512 MiB / 60 bodova

Županijsko natjecanje iz informatike 2019.

Prva podskupina (1. i 2. razred)

Pojašnjenje prvog primjera: Prvoga dana snijeg se nalazi na cijeloj planini i najduži spust sastoji se od polja s visinama: $9 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 1$ (duljina 5).

Nakon prvog dana snijeg će se otopiti na poljima visine 1, te je najduži spust drugoga dana: $9 \rightarrow 5 \rightarrow 4 \rightarrow 3$ (duljina 4).

Nakon četvrtog dana planina će izgledati kao na donjoj slici na kojoj su brojem 0 označena polja na kojima više nema snijega. Kao što vidimo na slici, petoga dana skijanja najdulji spust bit će $9 \rightarrow 5$ ili $8 \rightarrow 7$ (duljina 2).

0	0	0	5
7	8	0	9
0	0	0	0
5	0	0	0