

Školsko natjecanje iz informatike

Srednja škola

Druga podskupina (3. i 4. razred)

25. siječnja 2019.

ime zadatka	PRAVILAN	ŠPILJA	EPRUVETE
vremensko ograničenje	5 sekundi	5 sekundi	5 sekundi
broj bodova	40	50	60
	150		



Ministarstvo
znanosti i
obrazovanja



Agencija za odgoj i obrazovanje



HRVATSKA
ZAJEDNICA
TEHNIČKE
KULTURE



HRVATSKI SAVEZ
INFORMATIČARA

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.

Prikupljanje i evaluacija rješenja, objavljivanje liste rezultata te rješavanje eventualnih žalbi dužnost je članova školskog povjerenstva te ste dužni slijediti njihove upute prije, tijekom te nakon završetka natjecanja. Članovi školskih povjerenstava evaluirat će vaša rješenja koristeći unaprijed pripremljene test podatke. Za svaki zadatak trebate predati izvorni kod rješenja te (u slučaju korištenja programskog jezika C/C++) i odgovarajuću izvršnu (exe) datoteku. Radi lakše i brže evaluacije, imena datoteka moraju odgovarati imenima zadataka. Primjerice, ako se zadatak zove "Neboder", predajte datoteke neboder.py ili neboder.cpp i neboder.exe.

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. Vaši programi ne smiju pristupiti nikakvim datotekama ili ih kreirati.

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.exe < 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.

Primjeri pravilno napisanih programa

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

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

Izlaz: 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>

Za zadatke riješene u **Pythonu** potrebno je predati samo izvorni kod. Molimo da prva linija u kodu identificira inačicu Pythona koju treba koristiti kao u gornjim primjerima. Prilikom testiranja iz komandne linije potrebno je eksplicitno pozvati odgovarajući prevoditelj. Na primjer:

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

Vjerojatno vam je poznato da svaka riječ u sebi sadrži manje riječi, takozvane podriječi. Tako npr. riječ INFORMATIKA kao podriječi sadrži npr. INFO, MAT, FORMAT, te mnoge druge podriječi.

U zadanoj riječi možemo otkriti još više manjih riječi ako dopustimo „preskakanje“ nekih slova, tj. ako dopustimo da slova manje riječi ne budu nužno uzastopna u velikoj riječi. Na taj način u riječi INFORMATIKA možemo „pročitati“ npr. riječi IRMA ili FOTKA.

U ovom zadatku promatrat ćemo riječi koje možemo dobiti čitanjem nekih slova zadane riječi (s lijeva na desno) uz dopušteno preskakanje, ali tako da pročitana slova budu **pravilno razmaknuta**. Preciznije, kažemo da je riječ X **pravilan podniz** riječi Y ako je riječ X sadržana u riječi Y tako da se između slova riječi X nalazi uvijek **jednak broj preskočenih** slova riječi Y. Evo primjera:

- Riječ FRA **pravilan** je podniz riječi **INFORMATIKA** (između obilježenih slova F, R, A nalazi se po jedno preskočeno slovo). S druge strane, riječi IRMA i FOTKA nisu pravilni podnizovi riječi INFORMATIKA.
- Riječ META **pravilan** je podniz riječi **MATEMATIKA** (između obilježenih slova M, E, T, A nalaze se po dva preskočena slova).
- Naravno, možemo preskakati i nula slova, što znači da su obične podriječi (npr. FOR ili MAT u riječi INFORMATIKA) također pravilni podnizovi. To vrijedi i za jednoslovne podriječi, a tako je i svaka riječ **pravilan podniz** same sebe.

Napišite program koji unosi riječ i nekoliko „kandidata“ od kojih za svaki treba odgovoriti radi li se o pravilnom podnizu zadane riječi.

ULAZNI PODATCI

U prvom retku nalazi se riječ Y sastavljena od N ($3 \leq N \leq 20$) velikih slova engleske abecede.

U sljedećem retku nalazi se prirodan broj K ($2 \leq K \leq 6$), broj kandidata.

U svakom od sljedećih K redaka nalazi se kandidat – riječ sastavljena od barem jednog, a najviše N velikih slova engleske abecede

IZLAZNI PODATCI

Za svaku od K riječi (kandidata) u zaseban redak velikim slovima ispišite DA ili NE ovisno o tome radi li se o pravilnom podnizu riječi Y.

PRIMJERI TEST PODATAKA

ulaz

MATEMATIKA
6
MK
TATA
META
TEMA
MOTIKA
AEAIA

izlaz

DA
NE
DA
DA
NE
DA

Kamenu špilju prikazujemo tablicom od R redova i S stupaca, pri čemu je kamen predstavljen znakom '#' (hash ili ljestve), a prazan prostor znakom '.' (točka). Tlo špilje čini nakupina kamenja predstavljena **donjim** redovima tablice, a gornji dio („plafon“) čini čvrsta nakupina kamenja predstavljena **gornjim** redovima tablice, kao u sljedećem primjeru:

```
#####
####.##
#.#...#
.....
.....
..#...#
#####
```

Urušavanjem špilje cijeli gornji dio pada na donji, ne mijenjajući pritom svoj oblik: svi komadi u nakupini padaju povezani dok god je to moguće. Budući da je gornja nakupina vrlo čvrsta, čim neki kamen iz padajuće nakupine padne na kamen koji pripada tlu špilje, cijela nakupina prestane padati. Za gornji primjer nakon urušavanja dobivamo sljedeći izgled:

```
.....
.....
#####
####.##
#.#...#
..#...#
#####
```

Napišite program koji ispisuje izgled špilje nakon opisanog urušavanja.

ULAZNI PODATCI

U prvom retku nalaze se prirodni brojevi R i S ($3 \leq R, S \leq 10$) iz teksta zadatka. U sljedećih R redaka nalazi se po S znakova iz skupa {'.', '#'} koji opisuju izgled špilje kao što je opisano u tekstu zadatka. Gornji i donji redak sastojat će se samo od znakova '#' (kamen), a svaki kamen špilje bit će duž svog stupca povezan ili s gornjim, ili s donjim retkom špilje. Gornji i donji dio bit će odvojeni barem jednim retkom sastavljenim samo od znakova '.' (točka).

IZLAZNI PODATCI

Ispišite izgled špilje nakon urušavanja, u istom obliku i istih dimenzija kao u ulaznim podacima (R x S), koristeći iste znakove, pri čemu prostor oslobođen padanjem gornjeg dijela špilje ispunite točkama.

PRIMJERI TEST PODATAKA

ulaz	izlaz	ulaz	izlaz
5 8	8 5
#####	#####
.###..#.	#####	.###.	####.
.....	#####.	..#..	.###.
#...#...	######..
#####	#..
		..#..	.###.
		.###.	####.
		#####	

U svom laboratoriju Dubravka treba odmjeriti N mililitara otopine, ali ne može pronaći menzuru. Na raspolaganju su joj tri epruvete za koje zna da su im volumeni redom A mililitara, B mililitara i C mililitara, a u njima se već nalaze poznate količine otopine i to redom K mililitara, L mililitara i M mililitara. (To je sva otopina kojom raspolaže.)

U jednoj od tih epruveta Dubravka treba dobiti točno N mililitara otopine, a u tome će joj pomoći pretakanje. Jednim pretakanjem smatramo neku od sljedećih radnji:

- prelijevanje otopine iz jedne epruvete u drugu dok se prva ne isprazni ili druga ne napuni,
- izlijevanje sadržaja neke epruvete u otpad.

Evo primjera: pretpostavimo da treba dobiti 6 mL otopine koristeći epruvete volumena 3 mL, 5 mL i 8 mL ispunjene s 0 mL (prazna epruveta), 0 mL (prazna epruveta) i 8 mL (puna epruveta). Dovoljna su tri pretakanja:

1. iz treće epruvete napunimo drugu (5 mL), u trećoj ostaju 3 mL;
2. iz druge epruvete napunimo prvu (3 mL), u drugoj ostaju 2 mL;
3. sadržaj prve epruvete (3 mL) prelijemo u treću (3 mL) u kojoj tako dobivamo traženih 6 mL otopine.

Napišite program koji računa namanji broj pretakanja potreban da Dubravka obavi svoj zadatak.

ULAZNI PODATCI

U prvom retku nalazi se prirodan broj N ($1 \leq N \leq 20$), tražena količina otopine.

U drugom retku nalaze se prirodni brojevi A , B i C ($1 \leq A, B, C \leq 20$) odvojeni razmakom, volumeni epruveta od kojih će barem jedan biti veći ili jednak N .

U drugom retku nalaze se cijeli brojevi K ($0 \leq K \leq A$), L ($0 \leq L \leq B$) i M ($0 \leq M \leq C$) odvojeni razmakom, trenutne količine otopine u epruvetama, različite od N . U svim test podacima količine su takve da je Dubravkin zadatak moguće obaviti.

IZLAZNI PODATCI

U jedini redak ispišite traženi najmanji broj pretakanja.

PRIMJERI TEST PODATAKA

ulaz

6
3 5 8
0 0 8

izlaz

3

ulaz

4
3 5 8
0 0 8

izlaz

6

Pojašnjenje prvog primjera: vidi tekst zadatka.