



## Hrvatsko otvoreno natjecanje u informatici

2. kolo, 16. studenog 2019.

### Zadaci

Zadatak	Vremensko ograničenje	Memorijsko ograničenje	Bodovi
<b>Osijek</b>	1 sekunda	512 MiB	20
<b>Radio</b>	1 sekunda	512 MiB	30
<b>ACM</b>	1 sekunda	512 MiB	50
<b>Slagalica</b>	1 sekunda	512 MiB	70
<b>Checker</b>	1 sekunda	512 MiB	110
<b>Popcount</b>	1 sekunda	512 MiB	110
<b>Ukupno</b>			390



## Zadatak Osijek

Morao je doći i taj dan. Lega, vatreni navijač Osijeka, navijač koji nikad nije propustio nijednu Osijekovu prvenstvenu utakmicu, izgubio je dobar glas i po preporuci liječnika ne smije na veliki derbi. Zato Lega, umjesto da je u Gradskom Vrtu, sjedi ispred televizora i gleda utakmicu. Utjehu je pronašao u kutiji čipsa. Dok je u kutiji bilo čipsa, Lega bi izvadio točno  $K$  komada (ako ih je nema toliko, izvadio bi koliko ima), pojeo ih i šaptom uzviknuo: „Ajmo Bijelo-plavi!”.



Ako znamo da je u kutiji bilo  $N$  komada čipsa, koliko je puta Lega prošaputao navijački poklič?

### Ulazni podaci

U prvom je retku cijeli broj  $N$  ( $0 \leq N \leq 100$ ) iz teksta zadatka.

U drugom je retku prirodan broj  $K$  ( $1 \leq K \leq 100$ ) iz teksta zadatka.

### Izlazni podaci

U jedini redak ispišite traženi broj iz teksta zadatka

### Probni primjeri

ulaz

10

2

izlaz

5

ulaz

15

7

izlaz

3

ulaz

0

10

izlaz

0

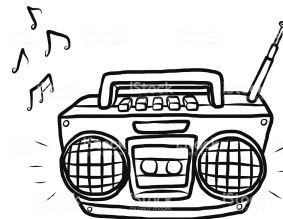
**Pojasnjenje prvog probnog primjera:** U kutiji je bilo 10 komada čipsa. Lega je 5 puta uzimao po 2 komada i uzvikivao navijački slogan.



## Zadatak Radio

Stjepan je, nakon godinu dana rada u poznatoj hrvatskoj firmi, kupio polovni BMW. Na putu od Zagreba do Belice (malog mjesta pored Pribislavca) Stjepan je glasno puštao glazbu i nakon **dugo** **dugo** trpljenja te buke njegova cura Ana mu je uputila  $N$  pritužbi.

Svaka pritužba je bila oblika: „Sjepane, molim te stišaj glazbu za  $A_i$  jedinica”. Stjepan bi rado poslušao svoju curu, ali također bi htio održati visoku razinu glasnoće glazbe. Kako bi pomirio svoje dvojbe Stjepan je odlučio povećati jačinu glazbe za  $B_i$  jedinica svaki put kad bi mu Ana uputila pritužbu, a zatim ispuniti želju svojoj djevojci i smanjiti jačinu glazbe za  $A_i$  jedinica.



Ako znamo da je jačina glazbe na početku puta bila  $X$  jedinica, pitamo se kolika je bila na kraju puta?

### Ulazni podaci

U prvom su retku dva prirodna broja  $N$  i  $X$  ( $1 \leq N, X \leq 10^5$ ) iz teksta zadatka.

U sljedećih su  $N$  redaka dva prirodna broja  $A_i$  i  $B_i$  ( $1 \leq A_i, B_i \leq 10^5$ ) iz teksta zadatka.

Jačina glazbe tokom puta nikada neće biti negativna.

### Izlazni podaci

U jedini redak ispišite jačinu glazbe na kraju puta.

### Bodovanje

U test podacima ukupno vrijednima 15 bodova vrijedit će da je  $N = 1$ .

### Probni primjeri

ulaz

1 10

1 5

izlaz

14

ulaz

2 7

4 1

3 1

izlaz

2

ulaz

3 500

400 200

100 50

100 20

izlaz

170

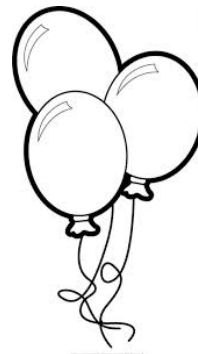
**Pojašnjenje prvog probnog primjera:** Na početku vožnje jačina glazbe je 10 jedinica. Nakon što Ana uputi prvu i jedinu pritužbu, Stjepan pojača radio za pet jedinica na 15, a zatim ga smanji za jednu jedinicu na 14.

**Pojašnjenje drugog probnog primjera:** Na početku vožnje jačina glazbe je 7 jedinica, nakon prve pritužbe jačina je 4 jedinice, a nakon druge i posljednje pritužbe jačina glazbe iznosi 2 jedinice.



## Zadatak ACM

Bliži se drevno (čitaj *drveno*) programersko natjecanje za djecu i mlade koje organizira nitko drugi doli ACM (*Avijatičarski Centar Metković*). Na natjecanju će nastupiti čak  $N$  timova uzrasta do šest godina. Među timovima je i zlatni trojac mladih avijatičara Hrvatske: Paula, Marin i Josip (obrnuto abecedno, svaka sličnost sa stvarnim događajima i osobama nije slučajna). Oblik natjecanja je standardan, dok kapetan posade radi piruete, kopilot čita zadatke na ruskom jeziku i Morseovom abecedom diktira kod programeru koji se nalazi izvan zrakoplova, ali je za njega sigurno pričvršćen ljepljivom vrpcom.



Timovi će na natjecanju ukrstiti koplja (preciznije krila) na  $M$  različitih zadataka. Timovi su na rang listi poredani silazno po broju riješenih zadataka.

- „Čekaj malo! Nisi objasnio ljudima kojim su redom poredani timovi sa istim brojem riješenih zadataka!” – dobacuje Marin kroz prozor svojeg krilatog ljubimca.
- „U pravu si, Marine.” – odgovorih mu.

Timovi koji imaju isti broj riješenih zadataka poredani su **uzlazno** po penalty vremenu. *Penalty* vrijeme za neki tim je suma penalty vremena svih točno riješenih zadataka. Penalty vrijeme točno riješenog zadatka odgovara vremenu zadnjeg poslanog rješenja za taj zadatak kojem se pridodaje **20 minuta** za svako pogrešno poslano rješenje. Tim neće slati rješenje za zadatak koji su već točno riješili. Najveći dozvoljen broj poslanih rješenja za isti zadatak je **20 po timu**. Ako timovi imaju isti broj riješenih zadataka i isto penalty vrijeme, poredani su po imenima abecedno.

Natjecanje traje **pet** sati. Tijekom prva četiri sata rang lista je vidljiva svim timovima te za svaki tim pokazuje informacije o svakom zadatku (koliko je ukupno slanja bilo, je li riješen i u koje vrijeme je riješen). Tijekom ta četiri sata, poredak na listi se sa svakim slanjem automatski ažurira. Nakon četiri sata, lista se zamrzne, tj. ostane u poretku u kojem je bila. Informacije o točnosti rješenja poslanih tijekom zadnjeg sata svaki tim zna samo za svoja vlastita, ali se za svaki tim i dalje za svaki zadatak na listi ažurira koliko je ukupno rješenja poslano i kada je poslano zadnje.

Natjecanje je završilo, lista će se uskoro odmrznuti, a naš trojac, tj. tim s imenom `NijeZivotJedanACM` treba vašu pomoć. Zanima ih koja je najniža moguća pozicija na kojoj mogu završiti kada se lista odmrzne. Ali u tih pet sati su se toliko izvrtjeli po modrom nebu da im je već zlo i programčić koji ovo provjerava nisu u stanju napisati sami. Pomozite im!

### Ulazni podaci

U prvom su retku prirodni brojevi  $N$  ( $1 \leq N \leq 1000$ ) i  $M$  ( $1 \leq M \leq 15$ ) iz teksta zadatka.

U sljedećih  $N$  redaka je stanje zamrznute liste na kraju natjecanja. Svaki red započinje imenom tima (riječ sastavljena od malih i velikih slova engleske abecede ne duža od 20 znakova, imena svih timova bit će različita) koje je razmakom odvojeno od  $M$  riječi koje su međusobno odvojene razmacima, a nose informacije o rješenjima zadataka za taj tim, redom od prvog do zadnjeg zadatka.

Riječi su za svaki zadatak oblika `SX/V`, gdje je:

- **S** stanje poslanih rješenja za taj zadatak (`+` označava da je zadatak točno riješen, `-` označava da nije, a `?` označava da je zadnje rješenje poslano nakon zamrzavanja ljestvice).
- **X** je ukupan broj rješenja koja je taj tim poslao za taj zadatak te se izostavlja ako je jednak nuli.
- **V** je vrijeme u kojem je poslano zadnje rješenje. Vrijeme je u formatu `HH:MM:SS` (sa vodećim nulama) te je manje od 5 sati. Cijeli `/V` dio se u riječi izostavlja ako zadatak nije točno riješen.

U posljednjem se retku nalazi se odmrznuti redak za naš trojac, tim s imenom `NijeZivotJedanACM`.



## Izlazni podaci

U prvi i jedini redak ispišite najnižu moguću poziciju na kojoj naš trojac može završiti nakon odmrzavanja liste.

## Bodovanje

U test podacima ukupno vrijednima 10 bodova, na ulazu se neće pojaviti znak '?'.  
U test podacima ukupno vrijednima dodatnih 15 bodova, svako poslano rješenje svakog tima bit će točno.

## Probni primjeri

ulaz

2 1

NijeZivotJedanACM -

ZivotJESTJedanACM -

NijeZivotJedanACM -

izlaz

1

ulaz

3 2

StoJeZivot ?1/04:00:00 +1/02:04:06

JeLiZivotJedanACM ?1/04:59:59 -

NijeZivotJedanACM ?1/04:42:43 -

NijeZivotJedanACM +1/04:42:43 -

izlaz

2

ulaz

7 4

NisamSadaNistaDonio +11/03:59:59 +3/03:42:02 +2/00:14:59 ?1/04:56:12

JeLiMojKockaSeUmio ?14/04:00:00 -13 +1/00:10:01 +19/03:04:42

OstaviDobroJe ?4/04:59:59 -1 +2/00:24:15 +18/03:24:45

DobroJeOstavi +1/01:42:53 - ?20/04:58:23 ?1/04:34:43

NijeZivotJedanACM ?2/04:50:05 ?14/04:32:12 +2/01:32:45 ?6/04:59:59

KoSeToSeta ?/04:23:32 - +10/01:00:00 -20

SipSipSipSipSipSip - - - ?20/04:00:00

NijeZivotJedanACM -2/04:50:05 +14/04:32:12 +2/01:32:45 +6/04:59:59

izlaz

3

### Pojašnjenje prvog probnog primjera:

Lista će nakon odmrzavanja biti ista kao i dok je bila zamrznuta, s našim trojcem na prvom mjestu!

### Pojašnjenje drugog probnog primjera:

U najgorem će slučaju naš trojac izgubiti samo od tima **StoJeZivot** i završiti na drugom mjestu.

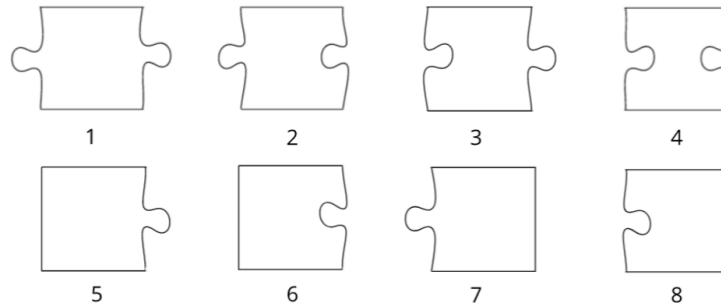
### Pojašnjenje trećeg probnog primjera:

U najgorem će slučaju naš trojac izgubiti od timova **NisamSadaNistaDonio** i **OstaviDobroJe** te završiti na trećem mjestu.



## Zadatak Slagalica

Mali Fabijan je za rođendan dobio jednodimenzionalnu slagalicu koja se sastoji od  $N$  komadića. Primijetio je da svaki komadić ima jedan od sljedećih oblika:



Dodatno, poznato je da se među tih  $N$  komadića nalazi točno jedan komadić oblika 5 ili 6 te točno jedan komadić oblika 7 ili 8.

Fabijan želi složiti sve komadiće u jednodimenzionalni slijed tako da prvi komadić u slijedu bude oblika 5 ili 6, a zadnji oblika 7 ili 8. Dva komadića može spojiti samo ako na rubu na kojem se dodiruju imaju suprotne oblike, dakle jedan ima udubinu, a drugi izbočinu.

Budući da mu je to prelagano, Fabijan je na svaki komadić napisao različit prirodan broj te se nakon toga zapitao kako bi trebao posložiti komadiće ako želi da niz što ga redom čine brojevi zapisani na komadićima nakon slaganja bude što manji. Niz  $A$  je manji od niza  $B$  ako za prvu poziciju  $i$  na kojoj se njihovi elementi razlikuju vrijedi  $A_i < B_i$ .

**Napomena:** Komadići se ne smiju rotirati.

### Ulazni podaci

U prvom je retku prirodan broj  $N$  ( $2 \leq N \leq 10^5$ ) iz teksta zadatka.

U sljedećih su  $N$  redaka dva prirodna broja  $X_i$  ( $1 \leq X_i \leq 8$ ) i  $A_i$  ( $1 \leq A_i \leq 10^9$ ), oblik  $i$ -tog komadića i broj koji je Fabijan napisao na njega. Brojevi  $A_i$  će biti međusobno različiti.

### Izlazni podaci

Ako Fabijan može složiti slagalicu, potrebno je ispisati redom brojeve na komadićima u složenoj slagalici koji tvore najmanji niz.

Ako Fabijan ne može složiti slagalicu, potrebno je ispisati  $-1$ .

### Bodovanje

U test podacima ukupno vrijednima 5 bodova vrijedi  $N \leq 10$ .

U test podacima vrijednima dodatnih 10 bodova neće se pojaviti komadići oblika 2 i 3.

U test podacima vrijednima dodatnih 20 bodova bit će najviše jedan komadić oblika 1 ili 4.

Ako za neki testni primjer ispišete ispravno slaganje, ali niz nije najmanji, ostvarit ćete 40% predviđenih bodova za taj test podatak.



## Probni primjeri

ulaz

5  
1 5  
2 7  
2 3  
8 4  
6 1

izlaz

1 3 7 5 4

ulaz

3  
5 1  
7 2  
4 3

izlaz

1 3 2

ulaz

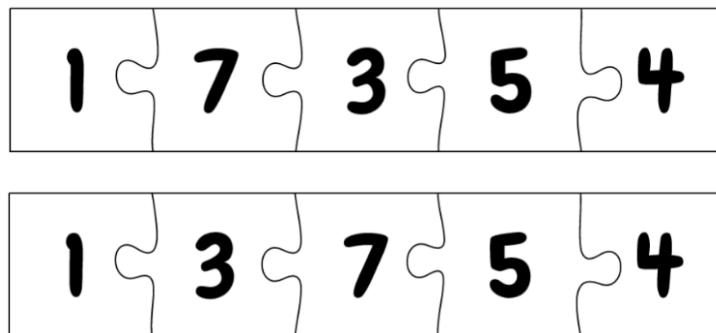
5  
2 5  
2 7  
2 3  
8 4  
6 1

izlaz

-1

### Pojašnjenje prvog probnog primjera:

Fabijan komadiće slagalice može složiti na dva različita načina:



Vidimo da drugo slaganje na drugom komadiću slagalice ima manji broj pa je to rješenje.



## Zadatak Checker

“...fool me once, shame on — shame on you. Fool me — you can't get fooled again.” – W.

U ovom zadatku promatramo pravilne  $N$ -terokute kojima su stranice obojene u tri boje, a vrhovi označeni prirodnim brojevima u smjeru kazaljke na satu. *Triangulacija* je podjela mnogokuta na trokute unutarnjim dijagonalama takva da dijagonale nemaju zajedničkih točaka osim vrhova mnogokuta te ne sijeku stranice mnogokuta osim u vrhovima mnogokuta. Naravno, u ovom zadatku i svaka će dijagonala biti obojena u jednu od tri boje.

Triangulacija je *domoljubna* ako za svaki od  $N - 2$  trokuta vrijedi da su mu sve tri stranice različite boje. Vaš je zadatak odrediti čine li dijagonale triangulaciju te je li ta triangulacija domoljubna.

### Ulazni podaci

U prvom je retku redni broj podzadatka kojem pripada testni primjer (vidi tablicu u poglavlju o bodovanju). Ako vaše rješenje ne mari za podzadatke, samo ga učitajte i ignorirajte.

U drugom je retku prirodan broj  $N$  iz teksta zadatka.

U trećem je retku  $N$ -teroznamenasti broj čije znamenke predstavljaju boje stranica  $N$ -terokuta u smjeru kazaljke na satu. Odnosno, prva znamenka predstavlja boju stranice  $(1, 2)$ , druga znamenka boju stranice  $(2, 3)$  i tako sve do  $N$ -te znamenke koja predstavlja boju stranice  $(N, 1)$ . Dakako, boje su označene znamenkama 1, 2 i 3.

U svakom od sljedećih  $N - 3$  redaka nalazi se po jedna dijagonala u obliku  $X Y C$ , gdje su  $X$  i  $Y$  vrhovi dijagonale, a  $C$  boja ( $1 \leq X, Y \leq N, 1 \leq C \leq 3$ ). Svaki redak će opisivati validnu dijagonalu, odnosno vrhovi  $X$  i  $Y$  neće biti ni isti ni susjedni.

### Izlazni podaci

Ako zadane dijagonale ne čine triangulaciju, ispišite **"neispravna triangulacija"** (bez navodnika).

Ako pak dijagonale čine triangulaciju, no ona nije domoljubna, ispišite **"neispravno bojenje"** (bez navodnika).

Ako dijagonale čine domoljubnu triangulaciju, ispišite **"tocno"** (bez navodnika).

### Bodovanje

Podzadatak	Broj bodova	Ograničenja
1	12	$4 \leq N \leq 300$
2	17	$4 \leq N \leq 2000$
3	23	$4 \leq N \leq 2 \cdot 10^5$ , odgovor je <b>neispravna triangulacija</b> ili <b>tocno</b>
4	23	$4 \leq N \leq 2 \cdot 10^5$ , odgovor je <b>neispravno bojenje</b> ili <b>tocno</b>
5	35	$4 \leq N \leq 2 \cdot 10^5$

Za razliku od zadatka *Trobojnica*, ako vaš program točno ispisuje prvi redak u svakom testnom primjeru nekog podzadatka, osvojit će 100% bodova predviđenih za taj podzadatak.





## Probni primjeri

**ulaz**

1  
5  
12113  
1 3 3  
2 5 2

**izlaz**

neispravna triangulacija

**ulaz**

1  
4  
1212  
1 3 2

**izlaz**

neispravno bojenje

**ulaz**

1  
7  
1223121  
1 3 3  
3 5 1  
5 7 3  
7 3 2

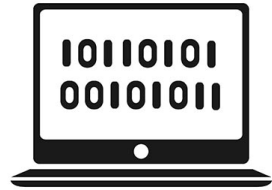
**izlaz**

tocno



## Zadatak Popcount

*Malo Algebarsko Napredno Računalo* (popularno MALNAR) posljednji je krik tehnologije u svijetu minijaturnih programabilnih računala. Vlastite programe za ovaj uređaj moguće je pisati u programskom jeziku *MalnarScript* kojeg krasi sljedeća obilježja:



- Ulaz u program je jedan nenegativan cijeli broj koji je strogo manji od  $2^N$ .
- Izlaz programa je jedan nenegativan cijeli broj koji je strogo manji od  $2^N$ .
- U izvornom kodu *MalnarScript*-a moguće je baratati točno jednom  $N$ -bitnom cjelobrojnom varijablom  $A$  bez predznaka (engl. *unsigned*). Na početku programa vrijednost varijable  $A$  odgovara ulazu u program, a njena završna vrijednost predstavlja izlaz programa.
- Izvorni kod sastoji se od najviše  $K$  naredbi oblika  $A = \langle \text{izraz} \rangle$  koje se slijedno izvršavaju te se svaka od njih mora sastojati od **najviše 1000 znakova**. Simbol  $\langle \text{izraz} \rangle$  definiramo rekursivno kao:

$$\langle \text{izraz} \rangle = A \mid \langle \text{broj} \rangle \mid (\langle \text{izraz} \rangle \langle \text{operator} \rangle \langle \text{izraz} \rangle)$$

Odnosno, simbol  $\langle \text{izraz} \rangle$  može biti varijabla  $A$ , može odgovarati simbolu  $\langle \text{broj} \rangle$  ili pak može unutar okruglih zagrada predstavljati dvočlani izraz u kojem se svaki operand pokorava istoj definiciji.

Simbol  $\langle \text{broj} \rangle$  u gornjoj definiciji predstavlja nenegativan cijeli broj u dekadskom zapisu koji je strogo manji od  $2^N$ , dok simbol  $\langle \text{operator} \rangle$  može biti  $+$ ,  $-$ ,  $|$ ,  $\&$ ,  $<<$  ili  $>>$  te redom može predstavljati operacije zbrajanja, oduzimanja, bitovnog ili, bitovnog i, posmaka ulijevo i posmaka udesno. Također, znak  $A$  se u izrazu smije pojavljivati **najviše 5 puta**.

U slučaju da prilikom zbrajanja ili oduzimanja dođe do aritmetičkog prelijevanja (engl. *overflow*) ili podlijevanja (engl. *underflow*), programski jezik *MalnarScript* će te operacije izvesti modulo  $2^N$ . Primjerice, uz pretpostavku  $N = 3$  izraz  $(7 + 3)$  u jeziku *MalnarScript* iznosi 2, a izraz  $(2 - 5)$  iznosi 5.

Desna strana jednakosti svake naredbe u konačnici se evaluira u jedan broj koji se tada sprema u varijablu  $A$ . Evaluacija samog izraza provodi se tako da se najprije svako pojavljivanje znaka  $A$  zamijeni sa trenutnom vrijednošću varijable  $A$ . Zatim se izraz računa kao u matematici, odnosno, najprije se rješavaju zagrade. Primijetite da je definicija ispravnog izraza takva da je prioritet operatora nebitan, odnosno, račun je u potpunosti definiran zagradama.

Vaš je zadatak napisati program koji ispisuje program u jeziku *MalnarScript* koji računa broj jedinica u binarnom zapisu ulaza u program.

### Ulazni podaci

U prvom su retku prirodni brojevi  $N$  i  $K$  ( $1 \leq N, K \leq 500$ ) iz teksta zadatka.

### Izlazni podaci

U prvom retku potrebno je ispisati broj naredbi traženog programa u jeziku *MalnarScript*.

U preostalim je retcima potrebno ispisati naredbe traženog programa. Sve naredbe moraju u potpunosti poštovati sintaksu jezika *MalnarScript* kako je opisano u tekstu zadatka.



## Bodovanje

Podzadatak	Broj bodova	Ograničenja
1	15	$1 \leq N \leq 100, K = N$
2	15	$N = 500, K = 100$
3	35	$1 \leq N \leq 32, K = 5$
4	45	$100 \leq N \leq 500, K = 10$

## Probni primjeri

ulaz

2 2

izlaz

1

$A = (A - ((A \& 2) \gg 1))$

ulaz

3 5

izlaz

2

$A = ((A \& 4) | ((A \& 3) - ((A \& 2) \gg 1)))$

$A = ((A \& 3) + ((A \& 4) \gg 2))$

### Pojašnjenje prvog probnog primjera:

$ulaz = 0 \Rightarrow izlaz = (0 - ((0 \& 2) \gg 1)) = (0 - (0 \gg 1)) = (0 - 0) = 0$

$ulaz = 1 \Rightarrow izlaz = (1 - ((1 \& 2) \gg 1)) = (1 - (0 \gg 1)) = (1 - 0) = 1$

$ulaz = 2 \Rightarrow izlaz = (2 - ((2 \& 2) \gg 1)) = (2 - (2 \gg 1)) = (2 - 1) = 1$

$ulaz = 3 \Rightarrow izlaz = (3 - ((3 \& 2) \gg 1)) = (3 - (2 \gg 1)) = (3 - 1) = 2$