

ZADATAK	DNA	PROCESI	XOR
izvorni kôd	dna.pas dna.c dna.cpp dna.cxx dna.py	procesi.pas procesi.c procesi.cpp procesi.cxx procesi.py	xor.pas xor.c xor.cpp xor.cxx xor.py
ulazni podaci	standardni ulaz		
izlazni podaci	standardni izlaz		
vremensko ograničenje	1 sekunda	3 sekunde	2 sekunde
memorijsko ograničenje	512 MB		
broj bodova	50	60	90
	200		



Agencija za odgoj i obrazovanje
Education and Teacher Training Agency



HRVATSKI SAVEZ
INFORMATIČARA



Ministarstvo znanosti,
obrazovanja i sporta

Svaka DNA molekula je lanac takozvanih *nukleotida*, a svaki *nukleotid* može biti jedan od četiri vrste: adenin (A), citozin (C), gvanin (G) i timin (T). Dakle jedan DNA lanac možemo predstaviti nizom znakova gdje je svaki znak veliko slovo A, C, G ili T.

Sekvenciranje DNA je postupak određivanja redoslijeda nukleotida u nepoznatoj molekuli DNA, a sekvenciranje dugačkih lanaca DNA je vrlo zahtjevan postupak. Jedna metoda sekvenciranja dugačkih molekula najprije napravi puno jednakih kopija te nepoznate molekule, zatim izloži sve te molekule zračenju koje ih sve nasumce razbije na kraće lance. Neki od tih kraćih lanaca se sekvenciraju te se pokušava donekle odrediti originalna molekula ‘slaganjem’ kraćih lanaca - takozvanih *blokova*. Pritom se često koristi informacija o nizovima za koje znamo da ih originalna DNA sigurno sadrži - takozvanim *uzorcima*.

```

      A C T G
        C T G G G
    Blokovi   T G G
                G A
                A C T G
    _____
    Uzorak   A C T G G G G A C T G
    
```

Slika 1: U podzadatku A blokovi se mogu preklapati, potrebno je izgraditi lanac jednak uzorku

Podzadatak A: Zadani su DNA lanci A_1, A_2, \dots, A_N koje nazivamo *blokovi*. Zadani su DNA lanci U_1, U_2, \dots, U_M koji nazivamo *uzorci*. Potrebno je za svaki uzorak odrediti da li ga je moguće dobiti spajanjem blokova. Prilikom gradnje uzorka možemo blokove spajati proizvoljnim redoslijedom, te svaki od blokova A_k možemo koristiti *nula, jednom ili više puta*. **Dozvoljeno je** da se blokovi prilikom spajanja preklapaju (ako imaju ista slova na pozicijama na kojima se preklapaju). Napišite program koji će, za svaki zadani uzorak, odrediti da li ga je moguće na ovaj način dobiti spajanjem zadanih blokova.

```

      C C A C T G
    Blokovi           G G
                       G G
                           C T A C G
    _____
    Uzorak   A C T G G G G C T
    
```

Slika 2: : U podzadatku B blokovi se ne smiju preklapati, potrebno je izgraditi lanac koji sadrži uzorak

Podzadatak B: Zadani su novi blokovi B_1, B_2, \dots, B_N . Zadan je jedan uzorak X . Potrebno je spajajući što je moguće manje blokova B_k napraviti DNA lanac Y koji sadrži (kao niz uzastopnih znakova) uzorak X . Prilikom gradnje lanca Y možemo blokove spajati proizvoljnim redoslijedom, te svaki od blokova B_k možemo koristiti *nula, jednom ili više puta*. **Nije dozvoljeno** da se blokovi prilikom gradnje lanca preklapaju. Napišite program koji će odrediti **koliko je najmanje blokova potrebno** da se napravi takav lanac te također određuje jedan način izgradnje lanca.

ULAZNI PODACI

U prvom redu nalazi se prirodni broj N ($1 \leq N \leq 100$) - broj zadanih blokova u svakom od podzadataka.

Sljedećih N redova sadrže blokove koji se koriste u podzadatku A. U K -tom od tih redova nalazi se blok A_K - niz od najviše 100 velikih slova.

U sljedećem redu nalazi se prirodni broj M ($1 \leq M \leq 20$) - broj zadanih uzoraka u podzadatku A. U K -tom od sljedećih M redova nalazi se uzorak U_K - niz od najviše 100 velikih slova.

Sljedećih N redova sadrže blokove koji se koriste u podzadatku B. U K -tom od tih redova nalazi se blok B_K - niz od najviše 100 velikih slova.

U sljedećem redu nalazi se uzorak X - zadani uzorak u podzadatku B, također niz od najviše 100 velikih slova.

Svi nizovi se sastoje od barem jednog slova, te se u njima pojavljuju se samo slova A, C, G ili T.

IZLAZNI PODACI

U prvi red ispišite rješenje podzadatka A - niz od točno M znakova '0' ili '1'. K -ti po redu znak u nizu treba biti '1' ako je moguće dobiti točno uzorak U_K spajanjem blokova uz dozvoljena preklapanja, a '0' inače.

U sljedeća dva reda ispišite rješenje podzadatka B.

U drugi red potrebno je ispisati traženi najmanji mogući broj blokova S potrebnih da se od blokova (bez preklapanja) izgradi neki lanac Y koji sadrži uzorak X .

U treći red potrebno je ispisati na koji način se Y može dobiti spajanjem S blokova. Točnije potrebno je ispisati M blokova redom kojim čine Y , odvojene s točno jednim znakom '|' (vertikalna crta, ASCII 124) bez razmaka. Svaki od blokova mora biti jedan od blokova A_K iz ulaza.

Napomena: Rješenje podzadatka B će uvijek postojati, ali ne mora biti jedinstveno.

BODOVANJE

U test podacima vrijednim 40% bodova vrijedi $N \leq 10$ te je dužina svih blokova i uzoraka najviše 10.

Ukoliko je prvi red ispravan natjecatelj dobiva 40% bodova za odgovarajući test podatak.

Ukoliko je drugi red ispravan natjecatelj dobiva 40% bodova za odgovarajući test podatak.

Ukoliko je treći red ispravan natjecatelj dobiva 20% bodova za odgovarajući test podatak.

Da bi dobili bodove za drugi ili treći red potrebno je da prvi red postoji i sadrži jedan niz od jednog ili više znakova bez razmaka.

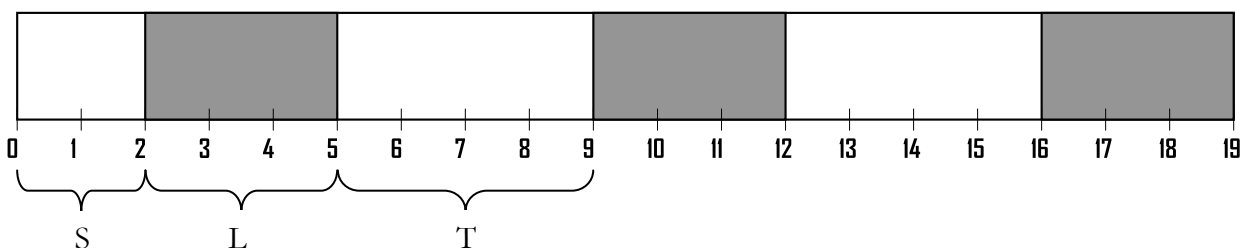
PRIMJERI TEST PODATAKA

ulaz 4 ACTG CTGGG TGG GA 3 ACTGGGGACTG GACTGG TGGC CCACTG GG CTACG TTCA ACTGGGGGCT izlaz 110 4 CCACTG GG GG CTACG	ulaz 1 ACCTG 4 ACCTG AACCTG ACCT ACCTGACCTG GGGTCCGTAAAAAAAAAAGTC AAAAA izlaz 1001 1 GGGTCCGTAAAAAAAAAAGTC
---	---

Raspoređivač (engl. scheduler) je dio jezgre operativnog sustava čiji je zadatak da odredi u kojem se trenutku koji proces izvodi na procesoru. Na raspolaganju imamo računalo s jednim procesorom, a vrijeme od početka rada procesora mjerimo u mikrosekundama. Svi procesi su periodične prirode, odnosno svaki se treba izvesti određen broj puta. Točnije, svaki proces K se izvodi na sljedeći način:

- Proces se prvi put pokreće u mikrosekundi S_K (točno S_K mikrosekundi nakon početka rada procesora).
- Proces se izvodi točno L_K mikrosekundi.
- Zatim dolazi pauza od točno T_K mikrosekundi.
- Odmah nakon pauze, proces se opet izvodi L_K mikrosekundi te se postupak ponavlja sve dok se proces ne izvede točno C_K puta.

Dakle proces će se izvoditi točno ukupno $C_K * T_K$ mikrosekundi, početak će u mikrosekundi S_K , potpuno završiti $C_K * L_K + (C_K - 1) * T_K$ mikrosekundi kasnije, a u tom vremenu bit će $C_K - 1$ perioda od T_K mikrosekundi u kojima se taj proces ne izvršava i kada se mogu izvršavati drugi procesi.



Slika 3: Proces s parametrima $S = 2$, $T = 4$, $L = 3$, $C = 3$

Raspoređivač je već odredio raspored izvršavanja nekih procesa. Raspored se sastoji od N procesa označenim brojevima od 1 do N , a proces K je opisan parametrima S_K , T_K , L_K , C_K kao gore. Raspored je takav da se može izvršiti na procesoru odnosno takav da se u svakom trenutku izvršava točno jedan od procesa (dozvoljeno je da u istoj mikrosekundi jedan proces završi izvođenje, a drugi započne).

Osim N procesa za koje je raspored već fiksiran, na procesoru je potrebno izvršiti M dodatnih procesa. Za svaki novi proces su zadani parametri T , L i C kao gore, a raspoređivač treba za svaki od njih, onim redom kojim su zadani, odrediti **najmanje moguće početno vrijeme S** kako bi se taj proces mogao izvršiti. Nakon što raspoređivač odredi to najmanje početno vrijeme za prvi novi proces, on se raspoređuje tako da počinje u tom vremenu te se postupak ponavlja za drugi proces, i tako dalje. Još jednom, početno vrijeme svakog procesa mora biti tako odabrano da se niti jedna dva procesa (uključujući i do sada raspoređene nove procese) ne izvršavaju u isto vrijeme.

Napišite program koji će **odrediti početna vremena S za sve nove zadane procese**. Rad procesora počinje u nultoj mikrosekundi te je moguće da ta nulta mikrosekunda bude početno vrijeme nekog (već raspoređenog ili novog) procesa.

ULAZNI PODACI

U prvom redu nalazi se prirodni broj, N ($1 \leq N \leq 100$) - broj već raspoređenih procesa. U svakom od sljedećih N redova nalaze se po četiri cijela broja S , T , L i C ($0 \leq S \leq 10^{12}$, $1 \leq T, L \leq 10^{12}$, $1 \leq C \leq 50$) koji redom označavaju vrijeme početka izvođenja procesa, vrijeme između dva izvođenja, trajanje jednog izvođenja te broj ponavljanja kao što je opisano u tekstu zadatka.

U sljedećem redu nalazi se prirodni broj, M ($1 \leq M \leq 10$) - broj novih procesa koje treba rasporediti. U svakom od sljedećih M redova nalaze se po tri cijela broja T , L i C ($1 \leq T, L \leq 10^{12}$, $1 \leq C \leq 50$) koji redom označavaju vrijeme između dva izvođenja, trajanje jednog izvođenja, te broj ponavljanja.

Napomena: Ne mora vrijediti da je već raspoređenim procesima odabrano najmanje moguće početno vrijeme.

IZLAZNI PODACI

Izlaz se treba sastojati od M redova. U K -ti red izlaza potrebno je ispisati jedan nenegativan cijeli broj – najraniji mogući početak izvođenja K -tog novog procesa s ulaza.

Napomena: Preporučamo da za učitavanje, računanje i ispis rezultata koristite 64-bitni cjelobrojni tip podataka (int64 u Pascalu, long long u C/C++).

BODOVANJE

U test podacima vrijednim 20% bodova N , M i svaki parametar C (za postojeće i nove procese) će biti manji ili jednaki 10, a ostali parametri (svaki S , T i L) će biti manji ili jednaki 1000.

U dodatnim test podacima vrijednim 20% bodova N , M i svaki parametar C će biti manji ili jednaki 10.

PRIMJERI TEST PODATAKA

<pre>ulaz 2 1 4 3 3 12 6 2 2 3 3 2 2 7 1 3 19 2 2 izlaz 18 6 4</pre>	<pre>ulaz 2 0 1 3 5 15 10 1 2 2 11 1 3 1 2 2 izlaz 7 20</pre>
---	--

Flota od N svemirskih brodova označenih prirodnim brojevima od 1 do N luta galaksijom u potrazi za novim otkrićima. Svaki brod K ima fiksni komunikacijski kanal X_K -- nenegativni cijeli broj manji od 2^{30} .

Vrijeme potrebno da jedan brod pošalje poruku drugome ovisi o njihovim komunikacijskim kanalima. Ako je kanal prvog broda X , a drugog Y onda se vrijeme slanja poruke računa tako da se:

- X i Y prebace u binarni sustav.
- Izračuna se binarni broj Z kojemu je K -ta binarna znamenka s desna jednaka 1 ako i samo ako su K -te binarne znamenke s desna brojeva X i Y različite.
- Vrijednost broja Z je traženo vrijeme slanja poruke.

Drugim riječima, vrijeme slanja poruke od broda A do broda B je takozvani *bitwise XOR* komunikacijskih kanala X i Y te se može izračunati pomoću izraza $X \oplus Y$ u C-u, C++-u, i Python-u, odnosno $X \text{ xor } Y$ u Pascal-u. Ako je na primjer $X = 12$ (binarno 1100), a $Y = 9$ (binarno 1001) onda je vrijeme slanja poruke 5 (binarno 101).

Podzadatak A: Svaki svemirski brod želi odrediti njemu najbliži svemirski brod (tj. onaj do kojeg je vrijeme slanja poruke najmanje) kako bi ga mogao pozvati u slučaju opasnosti. Napiši program koji će za svaki svemirski brod A **odrediti jedan njemu najbliži brod**. Ukoliko za neki A postoji više najbližih brodova potrebno je ispisati bilo koji od njih.

Podzadatak B: Svemirski brod broj 1 želi poslati poruku svim drugim brodovima u floti. To se odvija kroz $N - 1$ korak, a u svakom koraku jedan brod koji već ima poruku šalje je nekom brodu koji je još nije dobio. Ukupno vrijeme postupka je suma vremena slanja svih pojedinih poruka. Vaš program treba odrediti **najmanje moguće ukupno vrijeme** potrebno da poruku dobiju svi brodovi u floti.

ULAZNI PODACI

U prvom redu nalazi se prirodni broj, N ($2 \leq N \leq 100\,000$) - broj svemirskih brodova. U K -tom od sljedećih N redova nalazi se cijeli broj X_K ($0 \leq X_K < 2^{30}$) - komunikacijski kanal svemirskog broda K . Komunikacijski kanali bit će međusobno različiti.

IZLAZNI PODACI

U prvi red potrebno je ispisati N prirodnih brojeva, gdje je K -ti redu broj oznaka najbližeg broda brodu K .

U drugi red potrebno je ispisati traženo najmanje moguće ukupno vrijeme.

Napomena: Preporučamo da koristite 64-bitni cjelobrojni tip podataka (`int64` u Pascalu, `long long` u C/C++).

BODOVANJE

U test podacima vrijednim 20% bodova vrijedi $N \leq 1000$.

Ukoliko je prvi red izlaza ispravan natjecatelj dobiva 20% bodova za taj test podatak, čak i ako drugi red ne postoji ili nije ispravan.

Ukoliko je drugi red izlaza ispravan natjecatelj dobiva 80% bodova za taj test podatak, čak i ako prvi red nije ispravan.

PRIMJERI TEST PODATAKA

<pre>ulaz 4 1 2 4 8 izlaz 2 1 1 1 17</pre>	<pre>ulaz 5 1 3 6 8 12 izlaz 2 1 2 5 4 20</pre>	<pre>ulaz 8 15 7 6 21 25 5 14 10 izlaz 7 3 2 5 4 2 1 7 44</pre>
---	--	--