Transformacija (transformacija)

Neka su zadana dva niza pozitivnih cijelih brojeva od N i M elemenata redom: $\mathrm{A_i}, (0 \leq \mathrm{i} < \mathit{N})$, $\mathrm{B_j}, (0 \leq \mathrm{j} < \mathit{M})$. Elementi u oba niza su poredani u (nestrogo) rastućem poretku, odnosno $\mathrm{A_{i-1}} \leq \mathrm{A_i}, (0 < \mathit{i} < \mathit{N})$, $\mathrm{B_{j-1}} \leq \mathrm{B_j}, (0 < \mathit{j} < \mathit{M})$.
Neka je definisana i transformacija nad nizom B na sljedeći način. U svakom trenutku poznat je trenutni element niza B. U odnosu na ovaj element dopuštene su sljedeće operacije:

1. Obriši trenutni element. Trenutni element će postati sljedeći iz niza B ako takav postoji, u suprotnom završi transformaciju.
2. Ubaci novi element prije trenutnog elementa. Trenutni element se neće promijeniti, ali treba voditi računa da je sada ubačen novi element ispred njega.
3. Preskoči trenutni element bez promjene. Trenutni element će postati sljedeći iz niza B ako takav postoji, u suprotnom završi transformaciju.

Navedenom transformacijom moguće je svaki niz B transformisati u niz A u najvise $\mathrm{N} + \mathrm{M}$ koraka, ali često i manje. Bitno je naglasiti da se ispravnom transformacijom smatra samo ona koja je završena u skladu s pravilima operacija navedenim iznad.
Na primjer, neka je $\mathrm{A} = \{1, 2, 3, 5, 7, 9\}$ i $\mathrm{B} = \{1, 2, 4, 7, 8, 9, 10\}$. Jedna transformacija niza B u niz A je sljedeća (trenutni element je boldiran i podvučen).

Početno stanje	**1**, 2, 4, 7, 8, 9, 10
Operacija	***Niz B nakon transformacije***
1. Preskoči trenutni element	1, **2**, 4, 7, 8, 9, 10
2. Preskoči trenutni element	1, 2, **4**, 7, 8, 9, 10
3. Obriši trenutni element	1, 2, **7**, 8, 9, 10
4. Ubaci 3 prije trenutnog elementa	1, 2, 3, **7**, 8, 9, 10
5. Ubaci 5 prije trenutnog elementa	1, 2, 3, 5, **7**, 8, 9, 10
6. Preskoči trenutni element	1, 2, 3, 5, 7, **8**, 9, 10
7. Obriši trenutni element	1, 2, 3, 5, 7, **9**, 10
8. Preskoči trenutni element	1, 2, 3, 5, 7, 9, **10**
9. Obriši trenutni element	1, 2, 3, 5, 7, 9

Ustvari, ovo je transformacija niza B u niz A sa najmanjim mogućim brojem operacija.

Zadatak

Vaš zadatak je da napišete funckiju *TransformisiNiz* koja računa opisanu transformaciju sa najmanjim brojem operacija. Funkcija prima pet argumenata. Prvi argument je pozitivan

cijeli broj N, drugi argument je sortiran niz A od N pozitivnih cijelih brojeva. Analogno, slijede broj M i niz B. Peti argument je niz T alociran sa $N + M$ cijelih brojeva u koji trebate upisati opis transformacije na sljedeći način. Svaka operacija upisuje se u jedan element niza, redom kojim su se izvodile počevši od prvog elementa niza. Operacija „Obriši trenutni element“ opisuje se cijelim brojem -1, operacija “Ubaci novi element prije trenutnog elementa” cijelim brojem koji se ubacuje, a operacija “Preskoči trenutni element bez promjene” se opisuje cijelim brojem 0. Funkcija treba da vrati broj operacija, odnosno broj elemenata upisanih u niz T. Ukoliko postoji više od jedne ispravne transformacije sa istim najmanjim brojem operacija, u niz T možete upisati bilo koju.

Podzadatak 1 (11 bodova): $1 \leq N \leq 20, 1 \leq \mathrm{A_i}, \mathrm{B_i} \leq 1.000$

Podzadatak 2 (12 bodova): $1 \leq N \leq 10.000, 1 \leq A_i, B_i \leq 2.000.000.000$

Podzadatak 3 (23 bodova): $1 \leq N \leq 15.000.000, 1 \leq A_i, B_i \leq 60.000.000$

Podzadatak 4 (35 bodova): $1 \leq N \leq 2.000.000, 1 \leq A_i, B_i \leq 2.000.000.000$

Podzadatak 5 (19 bodova): $1 \leq N \leq 15.000.000, 1 \leq A_i, B_i \leq 2.000.000.000$

Primjeri

Primjer 1

$TransformisiNiz(6, \{1, 2, 3, 5, 7, 9\}, 7, \{1, 2, 4, 7, 8, 9, 10\}, T) = 9$

$T = \{\mathbf{0}, \mathbf{0}, \mathbf{-1}, \mathbf{3}, \mathbf{5}, \mathbf{0}, \mathbf{-1}, \mathbf{0}, \mathbf{-1}\}$
Primjer je objašnjen u tabeli iznad. Primijetite da su tačni i odgovori:
$T = \{0, 0, 3, -1, 5, 0, -1, 0, -1\}$, odnosno $T = \{0, 0, 3, 5, -1, 0, -1, 0, -1\}$.

Primjer 2
$TransformisiNiz(4, \{1,2,2,3\}, 3, \{2,2,3\}, T) = 4$

$T = \{1, 0, 0, 0\}$.

Detalji implementacije

Sa servera za takmičenje možete preuzeti pripremljena okruženja (*transformacija_c.zip*, *transformacija_cpp.zip* ili *transformacija_pas.zip*) sa osnovnim fajlovima za *C*, *C++* i *Pascal*.

Ukoliko koristite *C* ili *C++*, napišite funkciju s prototipom:

int TransformisiNiz(**int** n, **int*** krajnji, **int** m, **int*** pocetni, **int*** transformacija)

u fajlu *transformacija.[c/cpp]*.

Ukoliko koristite *Pascal*, napišite funkciju sa prototipom:

function TransformisiNiz(n : **LongInt**; krajnji : **Array of LongInt**; m : **LongInt**; pocetni : **Array of LongInt**; **var** rezultat : **Array of LongInt**) : **LongInt;**

u fajlu *transformacija.pas*.

Vodite računa da su nizovi indeksirani počevši od 0.

Samo unutar ovog fajla treba da implementirate svoje rješenje. Pri tome smijete koristiti i druge pomoćne funkcije koje ste vi napisali, te standardna zaglavlja/biblioteke odabranog programskog jezika i funkcije iz ovih biblioteka. Ne smijete ni na koji način vršiti interakciju sa standardnim ulazom/izlazom niti sa bilo kojom datotekom.

U pripremljenom okruženju nalazi se fajl *grader.[c/cpp/pas]* koji testira ispravnost rada[1] funkcije koju ste napisali na javne testne primjere. Kada šaljete svoje rješenje, šaljete samo fajl *transformacija.[c/cpp/pas]*, dok komisija koristi svoj *grader.[c/cpp/pas]* koji nije javni. U skladu s tim, slobodni ste da modificirate *grader.[c/cpp/pas]* i prilagođavate ga svojim potrebama u svrhu testiranja na lokalnom računaru.

Ukoliko koristite *Code::Blocks* u pripremljenim okruženjima možete naći i odgovarajuće projekte sa podešenim parametrima za prevođenje. *Release build* u potpunosti odgovara parametrima za prevođenje koji su na serveru za takmičenje, dok *Debug build* ima isključene optimizacije i uključene simbole za debagiranje.

Ukoliko koristite *FreePascal IDE*, dovoljno je da pokrenete prevođenje fajla *grader.pas* dok je u istom folderu fajl *transformacija.pas*. Na serveru za takmičenje postavljeni su sljedeći parametri za prevođenje: -dEVAL -vw -XS -O2.

Ukoliko ne koristite *Code::Blocks*, odnosno *FreePascal IDE*, u okruženjima se nalaze i fajlovi *prevedi_[c/cpp/pas].sh* koje možete koristiti za prevođenje svojih programa, a koje pozivate iz terminala komandom sh *prevedi_[c/cpp/pas].sh* iz odgovarajućeg foldera.

Okruženje za testiranje

Okruženje za testiranje koje vam je dostupno, podatke čita sa standardnog ulaza. Na prvoj liniji ulaza nalaze se brojevi N i M razdvojeni razmakom. Na drugoj liniji se nalazi N elemenata niza A, a na trećoj liniji M elemenata niza B.

Za prvi primjer naveden iznad, ulazni podaci bi izgledali ovako:

6 7
1 2 3 5 7 9
1 2 4 7 8 9 10

[1] Fajl grader.[c/cpp/pas] koji je javno dostupan testira samo ispravnost bez postavljanja ograničenja na vrijeme izvršavanja i iskorištenu memoriju.

Za drugi primjer naveden iznad, ulazni podaci bi izgledali ovako:

4 3
1 2 2 3
2 2 3