



Asia-Pacific Informatics Olympiad 10 Mei 2008

TUGAS	BEADS	ROADS	DNA
masukan	standard input		
keluaran	interaksi	standard output	
batas waktu	2 detik	1 detik	1 detik
batas memori	256 MB	128 MB	128 MB
nilai	100	100	100
	300		

Bahasa	Petunjuk Kompilasi
C	<code>gcc -o abc abc.c -std=c99 -O2 -DCONTEST -s -static -lm</code>
C++	<code>g++ -o abc abc.cpp -O2 -DCONTEST -s -static</code>
Pascal	<code>fpc -O1 -XS -dCONTEST abc.pas</code>

Durasi: 5 jam

3 soal

Semua soal harus dicoba.

Manik-Manik

Profesor X baru-baru ini meluncurkan hasil ciptaan terbarunya: Penukar Manik Terhebat (UBS dari *Ultimate Bead Swapper*). Seperti yang tertera, alat ini dapat membuat sebuah untaian manik-manik lebih menarik dengan cara menukar sejumlah manik-manik!

UBS memiliki N ban berjalan (*conveyor belts*) yang diletakkan secara paralel dengan arah Utara-Selatan. N ban berjalan tersebut diberi nomor 1 hingga N dari kiri ke kanan. Setiap ban bergerak dari Utara ke Selatan dengan kecepatan yang sama. Ada M penukar (*swappers*) diletakkan di antara dua ban yang bersebelahan. Tidak ada dua penukar yang terletak sama jauhnya dari ujung Utara dari UBS. (Dengan kata lain, mereka dapat diurutkan secara total berdasarkan seberapa jauh mereka berada dari ujung Utara.) Penukar-penukar tersebut diberi nomor 1 hingga M dari Utara hingga Selatan. Gambar 1 menunjukkan sebuah UBS dilihat dari atas.

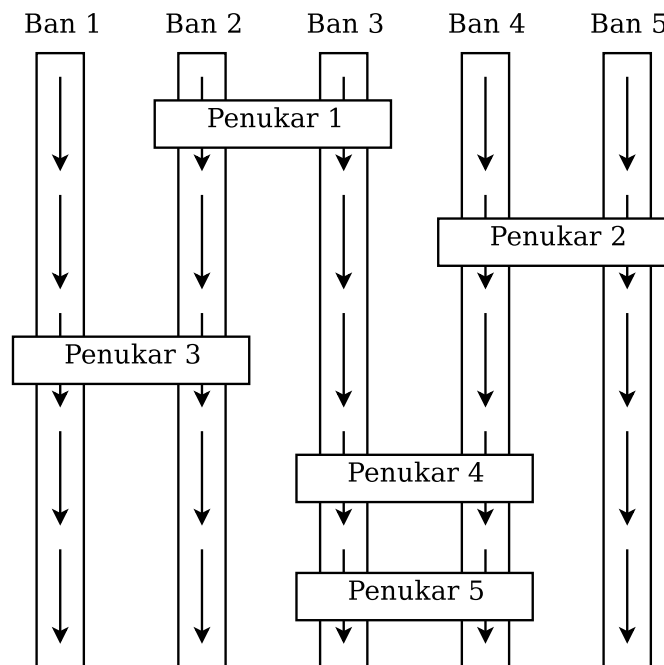


Figure 1: Sebuah Penukar Manik Terhebat dengan 5 ban berjalan dan 5 penukar.

Untuk menggunakan UBS, N manik-manik diletakkan di ujung utara dari ban berjalan pada saat yang sama sehingga mereka selalu membentuk sebuah baris horisontal di atas ban berjalan. Ketika dua buah manik terletak di bawah sebuah penukar, manik yang berada pada ban berjalan kanan berpindah posisi ke ban berjalan kiri, dan manik yang berada pada ban berjalan kiri berpindah posisi ke ban berjalan kanan. Setelah ditukar, kedua manik tidak merusak baris horisontal yang ada. Gambar 2 menggambarkan perilaku sebuah penukar.

Tugas

Tuliskan sebuah program yang ketika diberikan jumlah ban berjalan N , jumlah penukar M , dan posisi dari setiap penukar, menjawab pertanyaan dalam bentuk berikut:

Diberikan K dan J , untuk manik yang diletakkan pada ban K di ujung Utara dari UBS,

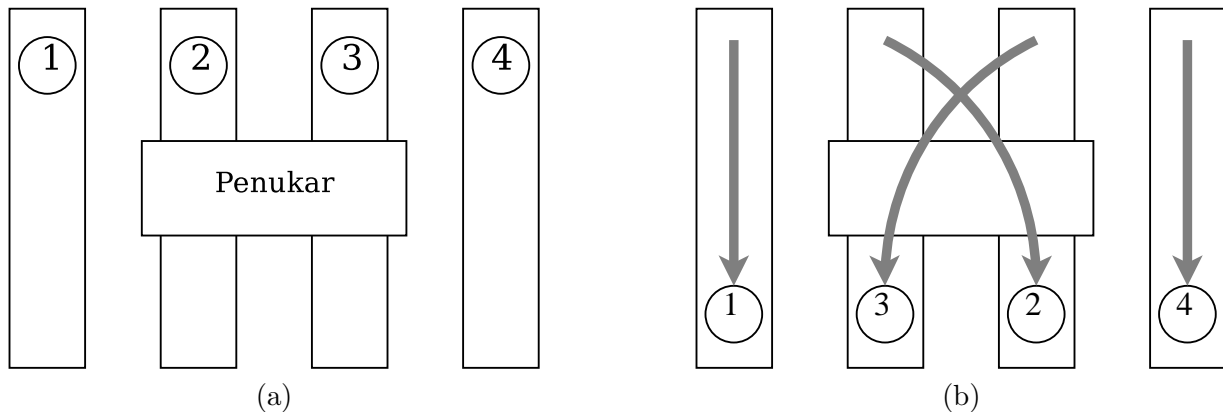


Figure 2: (a) Empat manik bergerak di atas ban berjalan. (b) Manik 2 dan 3 bertukar tempat setelah melalui sebuah penukar.

pada ban mana kah manik tersebut berada setelah semua manik bergerak melampaui penukar J ?

Masukan

Program Anda harus membaca dari *standard input*. Baris pertama berisi jumlah ban berjalan N ($1 \leq N \leq 300.000$) dan jumlah penukar M ($1 \leq M \leq 300.000$).

M baris berikutnya berisi lokasi setiap penukar, terurut dari Utara ke Selatan. Setiap baris berisi sebuah bilangan bulat P ($1 \leq P \leq M - 1$), yang berarti ada sebuah penukar yang menghubungkan ban berjalan P dan $P + 1$.

Interaksi

Setelah membaca masukan di atas, program Anda harus memanggil fungsi-fungsi yang tersedia dari perpustakaan (*library*) yang spesifikasinya dapat dilihat pada Tabel 1. Fungsi-fungsi tersebut harus dipanggil dalam urutan berikut ini:

1. Program memanggil fungsi `getNumQuestions` untuk mendapatkan Q ($1 \leq Q \leq 300.000$), jumlah pertanyaan yang akan ditanyakan.
2. Program memanggil sebanyak Q kali:
 - (a) fungsi `getQuestion` untuk menerima sebuah pertanyaan.
 - (b) fungsi `answer` untuk menjawab pertanyaan yang baru diterima.

Kami menekankan bahwa `getNumQuestions` harus dipanggil sekali saja pada saat pertama. `getQuestion` dan `answer` harus dipanggil bergiliran: setelah memanggil `getQuestion`, program Anda tidak boleh memanggil `getQuestion` sebelum program tersebut memanggil `answer`, and sebaliknya. Jika program Anda menyalahi konvensi ini ketika menjalankan sebuah tes, maka Anda akan mendapatkan skor 0% untuk tes tersebut.

Instruksi Pemrograman

Jika Anda mengumpulkan sebuah kode program Pascal, kode program tersebut harus berisi perintah berikut:

```
uses beadslib;
```

Jika Anda mengumpulkan sebuah program C atau C++, kode program tersebut harus berisi baris berikut:

```
#include "beadslib.h"
```

Prototipe Fungsi	Deskripsi
Pascal <code>function getNumQuestions():integer</code> C dan C++ <code>int getNumQuestions()</code>	Mengembalikan jumlah pertanyaan yang akan ditanyakan ke program Anda.
Pascal <code>procedure getQuestion(var K:integer, var J:integer)</code> C <code>void getQuestion(int *K, int *J)</code> C++ <code>void getQuestion(int &K, int &J)</code>	K adalah nomor ban berjalan manik yang dipertanyakan diletakkan di ujung Utara dari UBS. J ditetapkan sebagai nomor penukar.
Pascal <code>procedure answer(x:integer)</code> C dan C++ <code>void answer(int x)</code>	Melaporkan bahwa jawaban dari pertanyaan hasil pemanggilan <code>getQuestion</code> terakhir adalah <code>x</code> .

Table 1: *Library* interaksi.

Contoh *Library* dan Program

Anda akan diberikan sebuah berkas zip yang berisi kode program dari contoh *library* dan program. Berkas tersebut berisi tiga direktori — `pascal`, `c`, dan `cpp` — untuk kode program dalam Pascal, C, dan C++, secara berturut-turut. Setiap direktori berisi sebuah kode program dari contoh *library*, dan kode program dari sebuah program yang memanggil fungsi *library* dengan urutan yang benar.

Untuk Pascal, contoh *library* interaksi berada di unit `beadslib`, yang kode programnya dapat dilihat di `beadslib.pas`. Berkas `sample.pas` berisi sebuah kode program yang menggunakan *library* dengan benar.

Untuk C, contoh *library* interaksi berada di `beadslib.h`, yang spesifikasinya dapat dilihat di `beadslib.c`. Berkas `sample.c` berisi sebuah kode program yang menggunakan *library* dengan benar.

Untuk C++, contoh *library* interaksi juga berada di `beadslib.h` (tapi isi berkasnya tidak sama dengan untuk yang C), yang spesifikasinya dapat dilihat di `beadslib.cpp`. Berkas `sample.cpp` berisi sebuah kode program yang menggunakan *library* dengan benar.

Contoh *library* berperilaku sebagai berikut:

- Ketika `getNumQuestions` dari *library* contoh dipanggil, *library* tersebut membuka sebuah berkas `questions.txt`, membaca jumlah pertanyaan dan mengembalikan apa yang dibaca.
- Ketika `getQuestion` dipanggil, *library* membaca K dan J dari `questions.txt`.
- Ketika `answer` dipanggil, *library* mencetak argumen x ke *standard output*.
- *Library* mencetak sebuah pesan kesalahan ke *standard output* setiap kali sebuah fungsi dipanggil dengan urutan yang salah.

Berkas `questions.txt` memiliki bentuk sebagai berikut. Baris pertama berisi jumlah pertanyaan Q . Setiap baris dari Q baris berikutnya berisi dua bilangan bulat K , nomor dari sebuah ban berjalan, dan J , nomor dari sebuah penukar.

Contoh Masukan

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

Contoh isi `questions.txt`

```
2
3 4
5 5
```

(Masukan ini sesuai Gambar 1)

Contoh Interaksi

Pemanggilan Fungsi	Nilai hasil dan penjelasan
<code>getNumQuestions();</code>	2 Dua pertanyaan akan ditanyakan.
Pascal <code>getQuestion(K, J);</code> C <code>getQuestion(&K, &J);</code> C++ <code>getQuestion(K, J);</code>	$K=3, J=4$ Untuk sebuah manik yang berada di ban 3 dari ujung Utara dari UBS, pada ban mana kah manik tersebut berada setelah semua ban bergerak melewati penukar 4?
<code>answer(1);</code>	Setelah setiap manik melewati penukar 4, manik yang ditanyakan berada pada ban 1.
Pascal <code>getQuestion(K, J);</code> C <code>getQuestion(&K, &J);</code> C++ <code>getQuestion(K, J);</code>	$K=5, J=5$ Untuk sebuah manik yang berada pada ban dari ujung Utara dari UBS, pada ban mana kah manik tersebut berada setelah semua ban bergerak melewati penukar 5?
<code>answer(4);</code>	Setelah setiap manik melewati penukar 4, manik yang ditanyakan berada pada ban 1.



Batas Waktu dan Memori

Program Anda harus berhenti dalam 2 detik dan tidak menggunakan memori lebih dari 256 MB.

Penilaian

Nilai dari setiap skenario masukan adalah 100% jika program Anda menuruti konvensi pemanggilan fungsi di atas dan menjawab semua pertanyaan dengan tepat, dan 0% jika tidak.

Untuk sebuah kelompok tes bernilai 20 poin, M dan Q paling besar adalah 10.000.

Jalan

Kerajaan Asia Baru terdiri dari N desa yang terhubung oleh M jalan. Beberapa jalan terbuat dari batu-batuan, dan lainnya terbuat dari beton. Memelihara jalan-jalan bebas biaya membutuhkan biaya yang besar, dan tampaknya tidak mungkin untuk Kerajaan memelihara setiap jalan. Sebuah rencana pemeliharaan jalan yang baru dibutuhkan.

Misalnya, anggap desa-desa dan jalan-jalan di Asia Baru adalah seperti pada Gambar 1a. Jika Raja ingin dua jalan bebatuan bebas biaya, maka Kerajaan dapat menetapkan jalan (1,2), (2,3), (3,4), dan (3,5) bebas biaya seperti pada Gambar 1 b. Rencana ini memenuhi kriteria Raja karena (1) setiap dua desa dihubungkan oleh satu dan hanya satu jalan bebas, (2) ada sesedikit mungkin jalan bebas yang mungkin, dan (3) ada tepat dua jalan bebatuan: (2,3) dan (3,4).

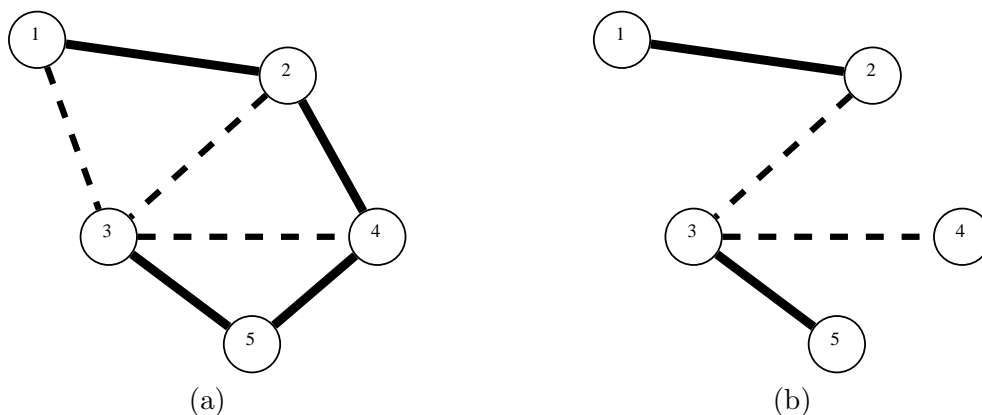


Figure 1: (a) Sebuah contoh konfigurasi desa dan jalan di Kerajaan Asia Baru. Garis tebal melambangkan jalan beton, dan garis putus-putus melambangkan garis bebatuan. (b) Sebuah rencana pemeliharaan jalan yang menghasilkan dua jalan bebatuan bebas biaya. Hanya jalan bebas biaya yang ditunjukkan.

Task

Diberikan deskripsi jalan-jalan di Asia Baru dan jumlah jalan bebatuan yang Raja ingin tetap gratis, tuliskan sebuah program yang menentukan jika ada sebuah rencana pemeliharaan jalan yang memenuhi kriteria Raja, dan keluarkan rencana yang valid jika ada.

Masukan

Baris pertama berisi tiga bilangan bulat dipisahkan oleh sebuah spasi:

- N , jumlah desa ($1 \leq N \leq 20.000$),
- M , jumlah jalan ($1 \leq M \leq 100.000$), dan
- K , jumlah jalan bebatuan Raja ingin tetap gratis ($0 \leq K \leq N - 1$).

M baris berikutnya mendeskripsikan jalan-jalan di Asia Baru, yang diberi nomor 1 hingga M . Baris ke- $(i + 1)$ menggambarkan jalan i . Setiap baris berisi tiga bilangan bulat dipisahkan spasi:

- u_i dan v_i , dua desa yang dihubungkan oleh Jalan i . Setiap desa diberi nomor 1 sampai dengan N , dan
- c_i , tipe jalan i ; $c_i = 0$ jika jalan i adalah jalan bebatuan, $c_i = 1$ jika jalan i terbuat dari beton.

Tidak ada lebih dari satu jalan yang menghubungkan sepasang desa.

Keluaran

Jika tidak ada rencana pemeliharaan jalan yang memenuhi kriteria Raja, program Anda harus mencetak `no solution` pada baris pertama dari keluaran.

Jika ada, maka program Anda harus mengeluarkan rencana pemeliharaan jalan yang valid dengan mendaftarkan jalan-jalan yang tetap bebas biaya, satu jalan per baris. Tulislah baris pada masukan yang menggambarkan sebuah jalan dalam daftar yang Anda buat. Jalan-jalan dapat didaftarkan dengan urutan sembarang. Jika ada lebih dari satu rencana yang valid, Anda dapat mengeluarkan rencana yang mana saja.

Contoh Masukan

```
5 7 2
1 3 0
4 5 1
3 2 0
5 3 1
4 3 0
1 2 1
4 2 1
```

(Masukan ini sesuai dengan Gambar 1a.)

Contoh Keluaran

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

(Keluaran ini sesuai dengan Gambar 1b.)

Batas Waktu dan Memori

Program Anda harus berhenti dalam 1 detik dan menggunakan memori tidak lebih dari 128 MB.

Penilaian

Nilai untuk sebuah skenario masukan adalah 100% jika jawaban yang benar dikeluarkan, dan 0% jika tidak.

Untuk sekelompok skenario tes bernilai 20 poin, K bernilai paling besar 10.

DNA

Salah satu penggunaan komputer yang menarik adalah untuk menganalisa data biologis seperti sekuens DNA. Secara biologis, sebuah rantai DNA adalah sebuah untai nukleotida Adenin, Sitosin, Guanin, dan Timin. Keempat nukleotida direpresentasikan menggunakan karakter A, C, G, dan T. Dengan demikian, sebuah untai DNA dapat direpresentasikan dalam sebuah *string* dari empat karakter ini. Kita menamakan *string* tersebut sebagai sebuah *sekuens DNA*.

Mungkin saja para ahli biologi tidak dapat menentukan sejumlah nukleotida dari sebuah rantai DNA. Dalam kasus tersebut, karakter N digunakan untuk mewakili nukleotida yang tidak diketahui pada sekuens DNA dari rantai tersebut. Dengan kata lain N adalah karakter kartu liar (*wildcard*) dari salah satu karakter dari A, C, G atau T. Kita namakan sebuah sekuens DNA berisi satu atau lebih karakter N *sekuens tidak lengkap* (*incomplete sequence*); selain itu, sekuens tersebut disebut *sekuens lengkap* (*complete sequence*). Sebuah sekuens lengkap dikatakan *setuju dengan* sebuah sekuens tidak lengkap jika sekuens lengkap tersebut adalah hasil dari mengganti setiap N di sekuens tidak lengkap tersebut dengan satu dari keempat nukleotida. Misalnya ACCCT setuju dengan ACNNT, tapi AGGAT tidak.

Para peneliti pada umumnya mengurutkan keempat nukleotida berdasarkan urutan abjad Latin A sebelum C, C sebelum G, G sebelum T. Sebuah sekuens DNA diklasifikasikan sebagai *bentuk-1* jika setiap nukleotida pada sekuens tersebut sama dengan atau muncul sebelum pada urutan abjad dengan nukleotida tepat di sebelah kanannya. Misalnya, AACCGT adalah bentuk-1, tapi AACGTC tidak.

Pada umumnya, sebuah sekuens adalah *bentuk-J*, untuk $J > 1$, jika sekuens tersebut adalah bentuk- $(j - 1)$ atau jika sekuens tersebut merupakan gabungan (*concatenation*) dari sekuens bentuk- $(j - 1)$ dengan sekuens bentuk-1. Misalnya, AACCC, ACACC, dan ACACA adalah bentuk-3, tapi GCACAC dan ACACACA tidak.

Para peneliti juga mengurutkan sekuens DNA secara leksikografis, sama seperti cara kita mengurutkan kata-kata pada kamus. Dengan demikian, sekuens bentuk-3 pertama dengan panjang 5 adalah AAAAA dan yang terakhir adalah TTTTT. Sebagai contoh lain, perhatikan sekuens tidak lengkap ACANNCNNG. Tujuh sekuens bentuk-3 pertama yang setuju dengan sekuens tidak lengkap tersebut adalah sebagai berikut:

```
ACAAACAAG
ACAAACACG
ACAAACAGG
ACAAACCAG
ACAAACCCG
ACAAACCCG
ACAAACCCG
ACAAACCTG
```

Tugas

Tulislah sebuah program untuk mencari sekuens bentuk- K ke- R yang setuju dengan sekuens tidak lengkap dengan panjang M yang diberikan.

Masukan

Baris pertama dari masukan berisi tiga bilangan bulat dipisahkan oleh sebuah spasi: M ($1 \leq M \leq 50.000$), K ($1 \leq K \leq 10$), dan R ($1 \leq R \leq 2 \times 10^{12}$). Baris kedua berisi sebuah *string* dengan panjang

M , yang merupakan sebuah sekuens tidak lengkap. Dijamin bahwa banyaknya sekuens bentuk K yang setuju dengan sekuens tidak lengkap tersebut tidak lebih dari 4×10^{18} , sehingga dapat direpresentasikan menggunakan tipe data `long long` pada C dan C++ atau `Int64` pada Pascal. Terlebih lagi, R tidak melebihi jumlah sekuens bentuk- K yang setuju dengan sekuens tidak lengkap yang diberikan.

Keluaran

Pada baris pertama, cetak sekuens bentuk- K ke R yang setuju dengan sekuens tidak lengkap pada masukan.

Contoh Masukan 1

```
9 3 5
ACANNCNNG
```

Contoh Keluaran 1

```
ACAAACCCG
```

Contoh Masukan 2

```
5 4 10
ACANN
```

Contoh Keluaran 2

```
ACAGC
```

Catatan Pemrograman

Pada C dan C++, Anda sebaiknya menggunakan tipe data `long long`. Potongan kode berikut ini menunjukkan bagaimana cara membaca dan menulis sebuah variabel bertipe `long long` dari dan ke *standard input/output*.

```
long long a;
scanf("%lld",&a);
printf("%lld\n",a);
```

Pada Pascal, Anda sebaiknya menggunakan `Int64`. Tidak ada instruksi spesial yang dibutuhkan untuk memanipulasi data dengan tipe ini.

Batas Waktu dan Memori

Program Anda harus berhenti dalam 1 detik dan tidak menggunakan memori lebih dari 128 MB.

Penilaian

Skor untuk setiap skenario masukan adalah 100% jika jawaban yang benar dicetak, dan 0% jika tidak. Untuk sekelompok skenario bernilai 20 poin, M paling besar adalah 10.