



## Həbsxana

 0.3 sec.  256 MB

Alice və Bob haqsız yerə maksimum təhlükəsizlikli həbsxanaya məhkum olunublar. İndi onların qaçış planı hazırlamaları lazımdır. Bunun üçün onlar mümkün qədər səmərəli şəkildə ünsiyyət qura bilməlidirlər (xüsusilə, Alice hər gün Bob-a məlumat göndərməlidir). Lakin onlar görüşə bilmirlər və yalnız salfetlərin üzərinə yazılmış qeydlərlə məlumat mübadiləsi edə bilirlər. Hər gün Alice Bob-a 0 ilə  $N - 1$  arasında yeni bir ədəd göndərmək istəyir. Hər naharda Alice üç salfet alır və onların hər birinə 0 ilə  $M - 1$  arasında bir ədəd yazır (təkrar ola bilər) və onları oturacağına qoyur. Sonra onların düşməni, Atila, salfetlərdən birini məhv edir və qalan ikisini qarışdırır. Nəhayət, Bob iki salfeti tapır və üzərindəki ədədləri oxuyur. O, Alice-in göndərmək istədiyi ilkin ədədi düzgün şəkildə bərpa etməlidir. Salfetlərdə məkan məhduddur, buna görə  $M$  sabitdir. Lakin Alice və Bob məlumat ötürülməsini maksimuma çatdırmaq istəyirlər, buna görə də  $N$ -i mümkün qədər böyük seçməkdə azaddırlar. Alice və Bob-a  $N$ -i böyütmək üçün strategiya hazırlamaqda kömək edin.



## İcra Detalları

Bu ünsiyyət məsələsi olduğundan, proqramınız iki ayrı icraatda (biri Alice, biri Bob üçün) işləyəcək və onlar arasında heç bir birbaşa əlaqə olmayacaq. Siz üç funksiyanı reallaşdırmalısınız:

```
int setup(int m);
```

Bu funksiya həm Alice-in, həm də Bob-un proqram icrasının əvvəlində bir dəfə çağırılacaq. Ona  $M$  veriləcək və o, seçilmiş  $N$ -i qaytarmalıdır. Hər iki `setup` çağırışı eyni  $N$  qaytarmalıdır.

```
std::vector<int> encode(int a);
```

Bu Alice-in strategiyasını reallaşdırır. Burada  $A$  ( $0 \leq A < N$ ) kodlaşdırılacaq ədəd verilir və  $0 \leq W_i < M$  olmaqla üç ədəd  $W_1, W_2, W_3$  qaytarılmalıdır. Bu funksiya  $T$  dəfə çağırılacaq - hər gün üçün bir dəfə (dəyərlər təkrarlana bilər).

```
int decode(int x, int y);
```

Bu Bob-un strategiyasını reallaşdırır. Burada `encode` tərəfindən qaytarılmış üç ədədin ikisi (istənilən ardıcılıqla) verilir. Funksiya Alice-in göndərdiyi ilkin  $A$  dəyərini qaytarmalıdır. Bu funksiya da  $T$  dəfə çağırılacaq - `encode`-un  $T$  çağırışına uyğun olaraq. Bütün `encode` çağırışları bitdikdən sonra `decode` çağırışları başlayacaq.



## Məhdudiyyətlər

- $M \leq 4300$
- $T = 5000$



## Qiymətləndirmə

Hər subtask üçün balın hesablanması  $S$  əmsalından və ən kiçik  $N$  dəyərindən asılıdır. Hədəf dəyər  $N^*$  verilir və tam bal almaq üçün  $N \geq N^*$  olmalıdır:

- Əgər həlliniz hər hansı bir testdə uğursuz olarsa,  $S = 0$ .
- Əgər  $N \geq N^*$ , onda  $S = 1.0$ .
- Əgər  $N < N^*$ , onda  $S = \max \left( 0.35 \max \left( \frac{\log(N) - 0.985 \log(M)}{\log(N^*) - 0.985 \log(M)}, 0.0 \right)^{0.3} + 0.65 \left( \frac{N}{N^*} \right)^{2.4}, 0.01 \right)$ .



## Subtask-lar

Alt tapşırıq	Bal	$M$	$N^*$
1	10	700	82017
2	10	1100	202217
3	10	1500	375751
4	10	1900	602617
5	10	2300	882817
6	10	2700	1216351
7	10	3100	1603217
8	10	3500	2043417
9	10	3900	2536951
10	10	4300	3083817



## Nümunə

Tutaq ki,  $T = 5$ . Burada sadə bir kodlaşdırma sxemi göstərək: Alice 0 göndərmək istədikdə üç eyni ədəd, 1 göndərmək istədikdə isə üç fərqli ədəd yazır. Bob hər iki salfeti oxumaqla ilkin ədədi düzgün təyin edə bilər.



İcra	Funksiya çağırışı	Nəticə
Alice	setup(10)	2
Bob	setup(10)	2
Alice	encode(0)	{5,5,5}
Alice	encode(1)	{8,3,7}
Alice	encode(1)	{0,3,1}
Alice	encode(0)	{7,7,7}
Alice	encode(1)	{6,2,0}
Bob	decode(5,5)	0
Bob	decode(8,7)	1
Bob	decode(3,0)	1
Bob	decode(7,7)	0
Bob	decode(2,0)	1

### Nümunə qiymətləndirici

Bütün encode və decode çağırışları eyni icraatın daxilində olacaq. Üstəlik, setup yalnız bir dəfə çağırılacaq. Giriş yalnız bir tam ədəddən –  $M$ -dən ibarət olacaq. Daha sonra program setup-un qaytardığı  $N$ -i çap edəcək, ardınca  $T$  dəfə encode və decode çağırışları ediləcək. Əgər həll uğursuz olsa, xəta mesajı veriləcək.