

Xabar

Aisha va Basma ikki do'stlar. Aishada Basmaga jo'natish lozim bo'lgan S bit(nollar va birlar)dan iborat M xabar bor. Aisha Basmaga **paketlar** jo'natish orqali bo'glana oladi. Paket deb 0 dan 30 gacha raqamlangan 31 ta bitlar ketma-ketligiga aytiladi. Aisha Basmaga bir nechta paketlar jo'natish orqali M xabarni yetkazmoqchi.

Afsuski, Kleopatra Aisha va Basmani aloqasini buzdi va u paketlarni **zararlashi** mumkin. Ya'ni, har bir paketda Kleopatra aniq 15 ta indeksdagi bitlarni o'zgartirib qo'yishi mumkin. Aniqroq qilib aytganda, 0 va 1 lardan iborat uzunligi 31 bo'lgan C massiv bor, va uning elementlari quyidagi ma'noga ega:

- $C[i] = 1$ bo'lsa, i -indeksdagi bit Kleopatra tomonidan o'zgartirib qo'yilishi mumkin. Bu indeksdagi bit Kleopatra **nazorat qiladigan** bitlar deb ataymiz.
- $C[i] = 0$ bo'lsa, i -indeksdagi bit Kleopatra tomonidan o'zgartirilmaydi.

C massiv aniq 15 ta birlar va 16 ta nollardan iborat. M xabarni jo'natish davomida, paketdagi Kleopatra o'zgartirishi mumkin bo'lgan indekslar to'plami o'zgarishsiz qoladi. Aisha aniq qaysi 15 ta indeks Kleopatra nazorat qiladigan bitligini biladi. Basma esa faqat 15 ta bit Kleopatra tomonidan o'zgartirilishi mumkinligini biladi, ammo aynan qaysi indekslarligini bilmaydi.

A deb Aisha jo'natmoqchi bo'lgan paketni aytaylik (bu paketni **original paket** deymiz). Shuningdek, B deb Basma qabul qilib olgan paketni aytaylik (bu paketni **zararlangan paket** deymiz). Har bir $0 \leq i < 31$ bo'lgan i uchun:

- agar Kleopatra i -indeksdagi bitni nazorat qilmasa ($C[i] = 0$), Basma i -bitni Aisha qanday jo'natgan bo'lsa shundayligicha qabul qiladi ($B[i] = A[i]$).
- aks holda, agar Kleopatra i -bitni nazorat qiladigan bo'lsa ($C[i] = 1$), $B[i]$ ning qiymati Kleopatra tomonidan belgilanadi.

Har bir paketni jo'natishi bilanoq, Aisha zararlangan paket qanday bo'lishini biladi.

Aisha hamma paketni jo'natganidan so'ng, Basma hamma zararlangan paketlarni **jo'natilgan tartibi bo'yicha** qabul qiladi va bu paketlar orqali original xabar M ni qayta qurishi lozim.

Sizning vazifangiz Aishaga M xabarni jo'natish uchun shunaqangi strategiyani topish kerakki, bunda Basma M xabarni zararlangan paketlar orqali qayta tiklashi imkoni bo'lsin. Aniqroq aytganda, ikkita funktsiyani kodlashingiz lozim. Birinchi funktsiya, Aisha qilishi kerak bo'lgan amallarni bajaradi. Unga M xabar va C massiv beriladi va bu funktsiya Basmaga bir nechta

paketlar jo'natishi lozim. Ikkinchi funksiya Basma qilishi kerak bo'lgan amallarni bajaradi. Unga zararlangan paket beriladi va bu funksiya original xabarni tiklashi lozim.

Kod yozish detallari

Kodlashingiz kerak bo'lgan birinchi funksiya:

```
void send_message(std::vector<bool> M, std::vector<bool> C)
```

- M : Aisha Basmaga jo'natmoqchi bo'lgan uzunligi S bo'lgan xabarni ifodalovchi massiv.
- C : Kleopatra nazorat qiladigan bitlar indeksini ifodalovchi uzunligi 31 bo'lgan massiv.
- Bu funksiya har bir test uchun **ko'pi bilan 2100 marta** chaqirilishi mumkin.

Bu funksiya paketni jo'natish uchun quyidagi funksiyadan foydalanishi lozim:

```
std::vector<bool> send_packet(std::vector<bool> A)
```

- A : Aisha jo'natgan bitlarni ifodalovchi original paket (uzunligi 31 bo'lgan massiv)
- Bu funksiya Basma qabul qiladigan bitlarni ifodalovchi zararlangan paket B ni qaytaradi.
- Bu funksiya har bir `send_message` chaqiruvi ichida ko'pi bilan 100 marta chaqirilishi mumkin.

Kodlashingiz kerak bo'lgan ikkinchi funksiya quyidagicha:

```
std::vector<bool> receive_message(std::vector<std::vector<bool>> R)
```

- R : zararlangan paketlarni ifodalovchi massiv. Paketlar `send_message`ni bitta chaqiruvidagi Aisha tomonidan yuborilgan paketlardan olingan va Aisha **jo'natgan tartibda** berilgan. R ning har bir qiymati zararlangan paketni ifodalovchi uzunligi 31 bo'lgan massiv.
- Bu funksiya original xabar M ni ifodalovchi S bitli massivni qaytarishi lozim.
- Bu funksiya har bir test uchun **ko'p marta** chaqirilishi mumkin, ammo har bir `send_message` chaqiruvi uchun **aniq bir marta** chaqiriladi. `receive_messagega chaqiruv tartibi` `send_message` ga chaqiruv tartibi bilan mos kelmasligi mumkin.

Shuningdek, baholash tizimida `send_message` va `receive_message` funksiyalari **alohida dasturlarda** joylashgan.

Chegaralar

- $1 \leq S \leq 1024$
- C da aniq 31 ta element bor, bulardan 16 tasi 0ga va 15 tasi 1 ga teng.

Qism masalalar va baholash

Agar, `send_packets` funksiyaga chaqiruvlarning birortasi yuqoridagi qoidalarga amal qilmasa yoki `receive_message` qaytargan javoblarning birortasi noto'g'ri bo'lsa, o'sha test uchun ballingiz 0 bo'ladi.

Aks holda, aytaylik Q `send_messaging` barcha chaqiruvlari ichida `send_packet`ni chaqirishlar sonining maksimali bo'lsin. Shuningdek, X quyidagicha ifodalansin:

- 1, agar $Q \leq 66$ bo'lsa
- 0.95^{Q-66} , agar $66 < Q \leq 100$ bo'lsa
- 0, agar $100 < Q$ bo'lsa

U holda, ballar quyidagicha hisoblanadi:

Qism masala	Ball	Qo'shimcha chegaralar
1	$10 \cdot X$	$S \leq 64$
2	$90 \cdot X$	Qo'shimcha chegaralarsiz.

Yodda tutingki, basi hollarda, grader **adaptive** ishlashi mumkin. Bu degani, `send_packet` qaytaradigan qiymatlar uning kiruvchi qiymatlari va bu funksiya oldingi qaytargan qiymatlarga bog'liq bo'lishi mumkin.

Misol

Quyidagi funksiya chaqiruvini ko'raylik:

```
send_message([0, 1, 1, 0],  
             [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
             1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1])
```

Aisha Basmaga yuborishga urinadigan xabar $[0,1,1,0]$. 0 dan 15 gacha raqamlangan bitlarni Kleopatra tomonidan o'zgartirib bo'lmaydi, 16-dan 30-gacha indekslardagi bitlar esa Kleopatra tomonidan o'zgartirilishi mumkin.

Faqat bu misol uchun, aytaylik Kleopatraning harakatlari deterministik bo'lsin, ya'ni u o'zi nazorat qiladigan bitlarni ketma-ket galma-galdan 0 va 1 larga o'zgartirib chiqsin. Masalan, u o'zi nazorat qiladigan birinchi bitga(bu holda 16-indeks) 0 belgilasin, ikkinchi bitga(bu holda 17-indeks) 1 belgilasin, uchinchi bitga(bu holda 18-indeks) 0 belgilasin, va hokazo.

Aisha quyidagi kabi original xabardagi ikkita bitni bitta paketda jo'natishi mumkin: u birinchi bitni o'zi nazorat qiladigan birinchi 8 ta bit orqali jo'natadi va ikkinchi bitni o'zi nazorat qiladigan keyingi

8 ta bit orqali jo'natadi.

So'ng Aisha quyidagi paketni jo'natadi:

```
send_packet([0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

Yodda tutingki, Kleopatra oxirgi 15 ta bitni o'zgartirishi mumkin, shuning uchun Aisha ularga ixtiyoriy qiymat berishi mumkin, chunki ular o'zgarishi mumkin. Misol uchun keltirilgan Kleopatra strategiyasi bilan, funksiya quyidagicha qiymat qaytaradi: $[0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0]$.

Aisha so'ng M xabarni so'nggi ikkita bitini yana shu tarzda jo'natadi.

```
send_packet([1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

Misol uchun keltirilgan Kleopatra strategiyasi bilan, funksiya quyidagicha qiymat qaytaradi: $[1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0]$.

Aisha ko'proq paket jo'natishi mumkin, lekin jo'natmaydi.

So'ng grader quyidagi funksiyani chaqiradi.

```
receive_message([[0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1,
                  0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0],
                 [1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
                  0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0]])
```

Basma M xabarni quyidagicha qayta tiklaydi. U har bitta paketdan birinchi va oxirgi ikki marta ketma ket uchraydigan bitni oladi. Ya'ni, birinchi paketdan, u $[0, 1]$ ni va ikkinchi paketdan $[1, 0]$ oladi. Bularni birga qo'yganda, u $[0, 1, 1, 0]$ xabarni qayta tiklaydi, bu esa `receive_message`ni ushbu chaqiruvi uchun to'g'ri javob.

Shuni ko'rsatish mumkinki, berilgan Kleopatra strategiyasi va uzunligi 4 bo'lgan xabarlar orqali, bu usul M ni qiymati qanday bo'lishidan qat'iy nazar Basmaga xabarni qayta tiklay olishini kafolatlaydi. Ammo, bu umumiy hol uchun to'g'ri emsa.

Namunaviy Grader

Namunaviy grader adaptive emas. Buning o'rniga, Kleopatranning harakatlari deterministik va u yuqoridagi misol kabi o'zi nazorat qiladigan bitlarn ketma-ket galma-galdan 0 va 1 larga o'zgartirib chiqadi.

Kiritish formati: **Kiruvchi ma'lumotlarning birinchi qatori testlar soni T ni o'z ichiga oladi.** T testlar, har biri quyidagicha ko'rinishda beriladi:

```
S
M[0]  M[1]  ...  M[S-1]
C[0]  C[1]  ...  C[30]
```

Chiqarish formati: Namunaviy grader har bitta T ta testning natijalarini kiruvchi ma'lumotlarda berilgan tartibda quyidagi formatda chiqaradi:

```
K L
D[0]  D[1]  ...  D[L-1]
```

Bu yerda, K send_packet ga chaqiruvlar sonini bildiradi, D esa receive_message qaytaradigan xabarni, L esa uni uzunligini ifodalaydi.