

Мессеж

Айша, Басма хоёр сайн найзууд бөгөөд тэд хоорондоо захидлаар харилцдаг. Айшад Басма руу илгээх S битийн дараалал (тэг эсвэл нэг орсон) бүхий M мессеж бий. Айша Басма руу **багцууд** илгээж харилцдаг. Нэг багц нь 0 -ээс 30 хүртэл дугаарлагдсан 31 битийн дараалал юм. Айша Басма руу хэдэн багц байдлаар тухайн M мессежийг илгээхийг хүсч байна.

Харамсалтай нь Клеопатра Айша, Басма хоёрын харилцааг хорлож, багцуудыг **өөрчилж** чаджээ. Ингэхдээ Клеопатра багц бүрээс яг 15 битийг л өөрчилдөг. Тодруулбал, 31 урттай C массив байгаа бөгөөд элемент бүр нь 0 эсвэл 1 утгатай байна. Энэ нь:

- Хэрэв $C[i] = 1$ бол i дугаартай битийг Клеопатра өөрчилж болохыг илэрхийлнэ. Бид эдгээр дугаартай битүүдийг Клеопатра **хяндаг** битүүд гэж нэрлэдэг.
- Хэрэв $C[i] = 0$ бол i дугаартай битийг Клеопатра өөрчлөх боломжгүй болохыг илэрхийлнэ.

C массив нь яг 15 нэг ба 16 тэгийг агуулна. Мессеж илгээх үед Клеопатрагийн хянадаг битийн дугаарууд бүх багцуудад ижил байна. Клеопатра аль 15 битийг хянадагийг Айша маш сайн мэддэг. Басма Клеопатра 15 битийг хянадаг гэдгийг мэддэг ч яг ямар дугаартай битүүдийг хянахыг мэдэхгүй.

A нь Айшагийн илгээхээр шийдсэн багц (үүнийг **жинхэнэ багц** гэж нэрлэе) байг. B нь Басмагийн хүлээн авсан багц (үүнийг **өөрчилсөн багц** гэж нэрлэе) байг. $0 \leq i < 31$ байх i бүрийн хувьд:

- Хэрэв Клеопатра i ($C[i] = 0$) дугаартай битийг хянахгүй бол Басма Айшагийн илгээсэн i дугаартай битийг зөв хүлээн авна ($B[i] = A[i]$),
- Үгүй бол, хэрэв Клеопатра i ($C[i] = 1$) дугаартай битийг хянавал $B[i]$ -ийн утгыг Клеопатра шийднэ.

Багц бүрийг илгээсний дараа Айша өөрчилсөн багц бүрийг олж мэддэг. Айша бүх багцуудыг илгээсний дараа Басма өөрчлөгдсөн бүх багцуудыг **илгээсэн дарааллаар нь хүлээн авч**, анхны жинхэнэ M мессежийг гаргаж авах хэрэгтэй.

Таны даалгавар бол Айшаг Басма руу M мессеж илгээх стратегийг боловсруулж хэрэгжүүлэх явдал бөгөөд ингэснээр Басма өөрчлөгдсөн багцуудаас M мессежийг сэргээх боломжтой болно. Тодруулбал, та хоёр функцийг хэрэгжүүлэх хэрэгтэй. Эхний функц нь Айшагийн үйлдлийг гүйцэтгэдэг. Энэ нь M мессеж болон C массив өгөгдсөн үед Басма руу мессеж дамжуулахын тулд хэдэн багц илгээх ёстой. Хоёр дахь функц нь Басмагийн үйлдлийг

гүйцэтгэдэг. Энэ нь өөрчилсөн багцууд өгөгдсөн үед M анхны жинхэнэ мессежийг сэргээх ёстой.

Хэрэгжүүлэлтийн мэдээлэл

Таны хэрэгжүүлэх эхний функц бол:

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

- M : S урттай массив нь Айшагийн Басма руу илгээхийг хүссэн мессеж юм.
- C : Клеопатрагийн хянадаг битүүдийн дугаарыг харуулсан 31 урттай массив.
- Энэ функцийг тест бүрт **хамгийн ихдээ 2100 удаа** дуудаж болно.

Энэ функц нь багц илгээхийн тулд дараах функцийг дуудна:

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

- A : Айшагийн илгээсэн битүүдийг илэрхийлэх жинхэнэ багц (31 урттай массив)
- Энэ функц нь Басмагийн хүлээн авах битүүдийг илэрхийлэх өөрчлөгдсөн багц B -г буцаана.
- Энэ функцийг `send_message` функцийн дуудлага бүрт хамгийн ихдээ 100 удаа дуудаж болно.

Таны хэрэгжүүлэх хоёр дахь функц бол:

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

- R : өөрчлөгдсөн багцуудыг илэрхийлэх нэг массив. Багцууд нь `send_message` функцийн нэг дуудлагаар Айшагийн илгээсэн багцуудаас **Айшагийн илгээсэн дарааллаар** орж ирдэг. R массивын элемент бүр нь 31 урттай массив бөгөөд өөрчлөгдсөн багцийг илэрхийлнэ.
- Энэ функц нь M анхны мессежтэй тэнцүү S бит массивыг буцаана.
- Энэ функцийг тест бүрт **олон удаа** дуудаж болно, харгалзах `send_message` дуудлага болгонд **яг нэг удаа** дуудагдана. `receive_message` функцийг дуудах дараалал нь харгалзах `send_message` функцийн дуудлагын дараалалтай ижил байх албагүй.

Үнэлгээний системд `send_message` болон `receive_message` функцуудыг **хоёр тусдаа программд** дууддаг болохыг анхаарна уу.

Хязгаарлалтууд

- $1 \leq S \leq 1024$
- C нь яг 31 элементтэй, үүнээс 16 нь 0 утгатай бөгөөд 15 нь 1 утгатай байна.

Дэд бодлогууд ба оноо

Тестийн аль нэг тохиолдолд `send_packet` функцийн дуудлага нь дээр дурьдсан дүрмүүдтэй нийцэхгүй эсвэл `receive_message` функцийн аль нэг дуудлагын буцах утга буруу байвал таны шийдлийн оноо энэ тестийн тохиолдолд 0 байх болно.

Үгүй бол тестийн бүх тохиолдлуудад `send_message`-ийн бүх дуудлагуудад `send_packet` функцийн дуудлагын хамгийн их тоог Q гэж үзье. Мөн X нь дараах утгатай байг:

- 1, if $Q \leq 66$
- 0.95^{Q-66} , if $66 < Q \leq 100$

Тэгвэл оноо нь дараах байдлаар тооцогдоно:

Дэд бодлого	Оноо	Нэмэлт хязгаарлалт
1	$10 \cdot X$	$S \leq 64$
2	$90 \cdot X$	Нэмэлт хязгаарлалт байхгүй.

Зарим тохиолдолд `grader`-ийн ажиллагаа нь **дасан зохицох чадвартай** байдаг гэдгийг анхаарна уу. Энэ нь `send_packet`-ийн буцаах утгууд нь зөвхөн түүний оролтын аргуудтаас гадна энэ функцийн өмнөх дуудлагын оролт, буцаах утгууд, `grader`-ийн үүсгэсэн псевдо санамсаргүй тоо зэрэг бусад олон зүйлээс хамаарна гэсэн үг юм. Хэрэв та үүнийг хоёр удаа ажиллуулж, хоёуланд нь ижил багцуудыг илгээвэл тэдгээрт ижил өөрчлөлтүүдийг хийнэ гэсэн утгаараа `grader` нь **детерминист шинж чанартай** байдаг.

Жишээ

Дараах дуудлагыг авч үзье

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

Айшагийн Басма руу илгээх мессеж нь $[0, 1, 1, 0]$ байна. 0-ээс 15 хүртэлх дугаартай битүүдийг Клеопатра өөрчлөх боломжгүй, харин 16-аас 30 хүртэлх дугаартай битүүдийг Клеопатра өөрчилж болно.

Энэ жишээний хувьд Клеопатра өөрийн хянадаг дараалсан битүүдийг 0 ба 1-ээр ээлжлэн дүүргэдэг, өөрөөр хэлбэл тэр өөрийн хянадаг эхний дугаартаа 0 оноодог (манай тохиолдолд 16-р утга), хоёр дахь дугаарт 1 (17-р утга), гурав дахь дугаарт 0 (18-р утга) гэх мэт.

Айша анхны мессежээс хоёр битийг нэг багцаар дараах байдлаар илгээх шийдвэр гаргаж болно. Тэр өөрийн хянадаг эхний 8 дугаар дээр эхний битийг, дараагийн 8 дугаар дээр хоёр дахь битийг илгээх болно. Дараа нь Айша дараах багцыг илгээхээр сонгосон:

```
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])
```

Клеопатра сүүлийн 15 битүүдийг солих боломжтой тул Айша тэдгээрийг дур мэдэн тохируулах боломжтойг анхаарна уу, учир нь тэдгээрийг дарж бичиж болно. Клеопатрагийн таамагласан стратегийн дагуу функц нь дараахь зүйлийг буцаана:

[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].

Айша хоёр дахь багц дахь M -ийн сүүлийн хоёр битийг өмнөхтэй ижил аргаар илгээхээр шийдэв.

```
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])
```

Клеопатрагийн таамагласан стратегийн дагуу функц нь дараахь зүйлийг буцаана: [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].

Айша илүү олон багц илгээж болох ч тэр илгээхгүй байхаар шийдсэн.

Дараа нь grader дараах функцийг дуудлагыг хийнэ:

```
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, 1, 0, 0, 0, 0, 0, 0, 0, 0,
                  0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1]])
```

Басма M мессежийг дараах байдлаар сэргээнэ. Багц бүрээс тэрээр дараалан хоёр удаа тохиолддог эхний битийг, мөн дараалан хоёр удаа тохиолддог сүүлчийн битийг авдаг. Өөрөөр хэлбэл, эхний багцаас тэрээр [0,1] бит, хоёр дахь багцаас [1,0] бит авдаг. Тэдгээрийг нийлүүлснээр тэрээр [0,1,1,0] гэсэн мессежийг сэргээдэг бөгөөд энэ нь 'receive_message' дуудлагын буцах зөв утга юм.

Клеопатрагийн таамагласан стратеги болон 4 урттай мессежийн хувьд Басмагийн энэхүү арга нь C -ын үнэ цэнээс үл хамааран M -г зөв сэргээдэг болохыг харуулж байна. Гэсэн хэдий ч ерөнхий тохиолдолд энэ нь зөв биш юм.

Жишээ Grader

Жишээ Grader нь дасан зохицох чадваргүй. Үүний оронд Клеопатра дээрх жишээнд тайлбарласны дагуу өөрийн хянадаг дараалсан битүүдийг 0 ба 1 битээр ээлжлэн дүүргэдэг.

Оролтын формат: Оролтын эхний мөрөнд хувилбаруудын тоо болох T бүхэл тоо өгөгдөнө, дараа нь T хувилбарууд гарч ирнэ. Тэдгээр тус бүрийг дараах хэлбэрээр өгсөн болно.

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

Гаралтын формат: Жишээ Grader нь T хувилбар бүрийн үр дүнг дараах форматаар оролтод өгсөнтэй ижил дарааллаар бичнэ.

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

Энд K нь `send_packet` функцийг дуудсан тоо, D нь `receive_message`-ийн буцаасан мессеж, L нь түүний урт юм.