

Алис, Боб ба Хэлхээ

Киберланд хэлхээ байгууллага нь n гишүүнтэй. Гишүүн бүр өөрийн дуртай тоо(дуртай тоо нь ялгаатай биш байж болно) болон үл давхцах нэртэй байдаг.

Гишүүд хоорондоо илгээсэн захидлын нийт тоо m . Захидал бүр нь илгээгч болон хүлээн авагчтай байдаг бөгөөд захидлын агуулга нь илгээгчийн дуртай тоо байна.

Гишүүн бүрийн үр дүнгийн утга нь тэдний хүлээн авсан захидлуудын агуулгын(илгээгчийн дуртай тоо) нийлбэрийг тооцоолж, 65536 (өөрөөр хэлбэл 2^{16}) тоогоор модуль авсан утга юм.

Таны даалгавар бол бүх гишүүдийн үр дүнгийн утгыг тооцоолох явдал юм.

Гэвч нөхцөл байдал харагдаж байгаа шиг тийм энгийн биш байна. Тэд энэ асуудлыг арай илүү төвөгтэй аргаар шийдэхээр болжээ.

- Алис бүх гишүүдийг (нэр болон дуртай тоог) мэддэг боловч захидлуудын тухай юу ч мэдэхгүй. Тэр Хэлхээ рүү 10^5 -ээс ихгүй урттай хоёртын тэмдэгт мөр илгээх шаардлагатай.
- Боб бүх m захидлыг(илгээгч болон хүлээн авагчийн нэрийг) мэддэг боловч гишүүдийн тухай мэдээлэлгүй. Тэр Хэлхээ рүү 10^5 -ээс ихгүй урттай хоёртын тэмдэгт мөр илгээх шаардлагатай.
- Хэлхээ нь Алис, Боб хоёрын илгээсэн хоёртын тэмдэгт мөрийг хүлээн авч, дараа нь гаралт болгон $16n$ бит бүхий хоёртын тэмдэгт мөр үүсгэх боломжтой. Гэвч түүний тооцоолох чадвар нь хязгаарлагдмал тул Хэлхээ нь зөвхөн үндсэн логик үйлдлүүдийг (Тодруулбал AND, OR, NOT) гүйцэтгэх чадвартай.

Дараах хэсэгт бид хэлхээ хэрхэн ажилладаг талаар дэлгэрэнгүй танилцуулав.

Хэлхээний дэлгэрэнгүй мэдээлэл

"Үүд" нь хэлхээний үндсэн элемент юм. "Үүд" нь тэг эсвэл хоёр логик оролт ("Үүд"-ийн төрлөөс хамаарч), нэг логик гаралтаас бүрдэнэ. Оролтын "Үүд", тооцооллын "Үүд" гэсэн хоёр төрлийн "Үүд" байдаг.

- Оролтын "Үүд" нь оролтгүй бөгөөд Алис, Боб хоёрын илгээсэн хоёртын тэмдэгт мөрийн битүүдийг илэрхийлдэг.
 - 0-ээс $(l_A + l_B - 1)$ хүртэлх тоогоор дугаарлагдсан $l_A + l_B$ ширхэг оролтын "Үүд" байх бөгөөд энд l_A, l_B нь харгалзан Алис, Боб нарын илгээх тэмдэгт мөрийн урт юм.
 - $0 \leq i < l_A$, байх i -ийн хувьд i -р "Үүд"-ийн гаралт нь Алисаас ирсэн тэмдэгт мөрийн i -р бит юм;
 - $0 \leq i < l_B$ байх i -ийн хувьд $(i + l_A)$ -р "Үүд"-ийн гаралт нь Бобоос ирсэн тэмдэгт мөрийн i -р бит юм.
- Тооцооллын "Үүд" нь хоёр оролттой бөгөөд тооцооллын процессыг илэрхийлдэг.
 - Тооцооллын "Үүд"-ийн дугаар нь $(l_A + l_B)$ -с эхэлдэг.
 - Тооцооллын "Үүд" бүрийн хувьд та оролтын хувьд хамааралтай хоёр "Үүд"-ийн дугаар, үйлдлийн төрөл $p(0 \leq p \leq 15)$ оруулах ёстой.
 - Тойрог хамаарлаас урьдчилан сэргийлэхийн тулд хоёр хамааралтай "Үүд"-ийн дугаар нь тооцооллын "Үүд"-ийн дугаараас бага байх ёстой.
 - Хэрэв түүний хоёр хамааралтай "Үүд"-ийн гаралт нь x_0 ба x_1 ($x_0, x_1 \in \{0, 1\}$) байвал тооцооллын "Үүд"-ийн гаралт нь:

$$f(p, x_0, x_1) = \left\lfloor \frac{p}{2^{x_0+2x_1}} \right\rfloor \bmod 2$$

Танд хэрэгтэй байж болох зарим жишээг энд үзүүлэв:

x_0	x_1	x_0 AND x_1 $f(8, x_0, x_1)$	x_0 OR x_1 $f(14, x_0, x_1)$	x_0 XOR x_1 $f(6, x_0, x_1)$	NOT x_0 $f(5, x_0, x_1)$
0	0	0	0	0	1
1	0	0	1	1	0
0	1	0	1	1	1
1	1	1	1	0	0

Хэрэгжилтийн дэлгэрэнгүй мэдээлэл

Анхаарна уу:

- Бүх массивын индексүүд 0-с эхэлдэг. Жишээлбэл a нь n урттай массив бол $a[0]$ -ээс $a[n - 1]$ хүртэл өгөгдөл хадгалах бөгөөд энэ мужаас хэтэрсэн индекст хандах нь хязгаараас хэтэрсэн алдаа үүсгэнэ.
- Бүх тэмдэгт мөрүүд хоосон тэмдэгт("\0")-ээр төгсдөг.

Та дараах функцуудыг хэрэгжүүлэх хэрэгтэй:

Alice

```
int alice(const int n, const char names[][5], const unsigned short numbers[], bool outputs_alice[]);
```

Чиглэл	Утга	Урт	Тайлбар	Хязгаарлалт
Оролт	n	1	n	$0 \leq n \leq 700$
	names	n	Гишүүдийн нэр.	Бүх нэрс нь ялгаатай бөгөөд зөвхөн англи жижиг үсгээс бүрдэх ба хамгийн ихдээ 4 тэмдэгттэй байна.
	numbers	n	Гишүүдийн дурай тоо.	Тоо бүр нь 0-ээс 65535 хооронд байна.
Гаралт	outputs_alice	l_A	Хэлхээ рүү илгээх хоёртын тэмдэгт мөр.	
	(Буцах утга)	1	l_A	Та l_A нь 10^5 -ээс хэтрэхгүй эсэхийг шалгах хэрэгтэй бөгөөд n ижил байвал l_A -г засах шаардлагатай.

Bob

```
int bob(const int m, const char senders[][5], const char recipients[][5], bool outputs_bob[]);
```

Чиглэл	Утга	Урт	Тайлбар	Хязгаарлалт
Оролт	m	1	m	$0 \leq m \leq 1000$
	senders	m	Захидал илгээгчийн нэрс.	Бүх нэрүүд Алисын оролтод бий
	recipients	m	Захидал хүлээн авагчийн нэрс.	
Гаралт	outputs_bob	l_B	Хэлхээ рүү илгээх хоёртын тэмдэгт мөр.	
	(Буцах утга)	1	l_B	Та l_B нь 10^5 -ээс хэтрэхгүй эсэхийг шалгах хэрэгтэй бөгөөд m ижил байвал l_B -г засах шаардлагатай.

Circuit

Хэлхээний тооцооллын процесс нь ерөнхий хэлхээтэй адил байхын тулд та Алис, Боб хоёроос хэлхээ рүү илгээсэн хоёртын тэмдэгт мөрийг шууд авах боломжгүй. Та зөвхөн эдгээр хоёр тэмдэгт мөрийн уртыг мэдэж, хэлхээний бүтцийг гаргана.

```
int circuit(const int la, const int lb, int operations[], int operands[][2], int outputs_circuit[][16]);
```

Чиглэл	Утга	Урт	Тайлбар	Хязгаарлалт
Оролт	la	1	l_A	
	lb	1	l_B	
Гаралт	operations	l	Хэлхээний "Үүд" бүрийн гүйцэтгэх үйлдлийн төрөл.	0-ээс 15 хүртэлх бүхэл тоо.

Чиглэл	Утга	Урт	Тайлбар	Хязгаарлалт
	operands	l	Хэлхээний "Үүд" бүрийн ашиглах операнд.	Энэ тоо нь одоогийн "Үүд"-ийн дугаараас бага байх ёстой.
	outputs_circuit	n	Хэлхээний гаралтын "Үүд"-ийн дугаар.	$outputs_{circuit}[i][j]$ нь i -р гишүүний эцсийн үр дүнгийн утгын j -р битийг (хамгийн бага ач холбогдолтой битээс тоолох) илэрхийлнэ. Алисын оролтын дагуу гишүүдийг эрэмбэлсэн.
	(Буцах утга)	1	l , энэ нь нийт "Үүд"-ийн тоог илэрхийлдэг (оролтын "Үүд"-ийг оруулаад).	Та $l \leq 2 \times 10^7$ гэдгийг хангах хэрэгтэй.

Хэдийгээр та $l_A + l_B$ -ээс бага дугаартай "Үүд"-ийн мэдээллийг *operations* ба *operands* массивын хувьд өөрчлөх боломжтой ч шалгагч (grader) ийм өөрчлөлтийг үл хэрэгсэх болно.

Жишээ

Дараах дуудалтуудыг хийсэн гэж үзье:

```
alice(3, {"alic", "bob", "circ"}, {10000, 20000, 30000}, outputs_alice);
bob(5, {"alic", "bob", "bob", "circ", "circ"}, {"circ", "circ", "alic", "circ", "circ"}, outputs_bob);
```

Энэ нь дараах хувилбарыг илэрхийлж байна:

- Алис 3 гишүүн байдаг гэдгийг мэддэг, *alic* нэртэй гишүүний дуртай тоо нь 10000 байна гэх мэт. *alice()* функцийн боломжит гаралт нь:
 - alice()* функцийн буцаах утга нь 2 бөгөөд энэ нь $l_A = 2$ болохыг илэрхийлнэ.
 - alice()* функц рүү $outputs_{alice}[0] = 1, outputs_{alice}[1] = 0$ гэж дамжуулснаар, үр дүнгийн хоёртын тэмдэгт мөр нь 10 байна.
- Боб 5 захидал байгааг мэдэх бөгөөд эхний захидал нь *alic*-ээс *circ* руу илгээгдсэн гэх мэт. *bob()* функцийн боломжит гаралт нь:
 - bob()* функцийн буцаах утга нь 3 бөгөөд энэ нь $l_B = 3$ болохыг илэрхийлнэ.
 - bob()* функцрүү $outputs_{bob}[0] = 1, outputs_{bob}[1] = 1, outputs_{bob}[2] = 0$ гэж дамжуулснаар, үр дүнгийн хоёртын тэмдэгт мөр нь 110 байна.

alice() болон *bob()* функцуудын өмнөх гаралтууд дээр үндэслэн дараах дуудалт хийгдэнэ:

```
circuit(2, 3, operations, operands, outputs_circuit);
```

Энэ функцийн хувьд зөв гаралт нь:

- circuit()* функцийн буцаах утга нь 7 бөгөөд бид 5 ба 6 гэсэн дугаартай хоёр тооцооллын "Үүд" нэмнэ гэсэн үг.
- circuit()* функц рүү *operations*, *operands*, *outputs_circuit* массивуудыг дараах байдлаар дамжуулна:
 - operations* = -1, -1, -1, -1, -1, 8, 14, энд оролтын "Үүд"-ийн үл хэрэгсэх мэдээллийг илэрхийлэхийн тулд -1 утга ашигладаг;
 - operands* = -1, -1, -1, -1, -1, -1, -1, -1, -1, 0, 4, 2, 5;
 - outputs_circuit* = 5, 5, 5, 5, 5, 6, 5, 5, 6, 6, 5, 5, 6, 5, Энэ массив бага зэрэг урт тул та хавсралт дахь *abc.cpp* файлаас бүтэн массивыг үзээрэй.

Гаралтруу тооцоолол хийх функц нь:

- 0 дугаартай "Үүд" ба 4 дугаартай "Үүд"-ийн оролттой 8 төрлийн тооцооллын "Үүд" нэмнэ. 0 дугаартай "Үүд"-ийн гаралт нь Алисаас авсан тэмдэгт мөрийн 0 дахь бит бөгөөд энэ нь 1; 4 дугаартай "Үүд"-ийн гаралт нь Бобоос авсан тэмдэгт мөрийн 2 дугаар бит бөгөөд энэ нь 0 юм. Тэгэхээр 5 дугаартай "Үүд"-ийн гаралт нь $f(8, 0, 1) = 0 \text{ AND } 1 = 0$ байна.
- 2 дугаартай "Үүд", 5 дугаартай "Үүд"-ийн оролттой 14 төрлийн тооцооллын "Үүд" нэмнэ. 2 дугаартай "Үүд"-ийн гаралт нь Бобоос ирсэн тэмдэгт мөрийн 0 дэх бит бөгөөд энэ нь 1; 5 дугаартай "Үүд"-ийн гаралт 0 байна. Тэгэхээр 6 дугаартай "Үүд"-ийн гаралт нь $f(14, 1, 0) = 1 \text{ OR } 0 = 1$ байна.
- $output_{circuit}[0]$ нь *alic*-ын эцсийн үр дүнгийн утгыг илэрхийлдэг бөгөөд энэ нь $(0100111000100000)_2 = 20000$ байна. *alic* зөвхөн *bob*-оос захидал хүлээн авсан тул *alic*-ын эцсийн үр дүнгийн утга нь 20000 байна.
- &bob-ын эцсийн үр дүнгийн утга нь 0 байх ёстой, учир нь тэр захидал хүлээн аваагүй; *circ*-ын эцсийн үр дүнгийн утга нь $(10000 + 20000 + 30000 + 30000) \bmod 65536 = 24464$ байх ёстой.

Хавсралт дахь *abc.cpp* файл нь энэ жишээг шалгаж болох боловч бусад тестийн тохиолдолд ажиллаж чадна гэж бид баталгаа өгөхгүй.

Хязгаарлалтууд

Бүх тестийн хувьд:

- $0 \leq n \leq 700, 0 \leq m \leq 1000$.
- Бүх нэрс нь ялгаатай бөгөөд зөвхөн англи жижиг үсгээс бүрдэх ба хамгийн ихдээ 4 тэмдэгттэй байна.
- Гишүүдийн дуртай тоо нь 0-ээс 65535 хооронд утгатай.
- Бүх илгээгч болон хүлээн авагчдын нэрс Алисын *names* оролтын массивт байна.
- *alice()* болон *bob()* функцийг санах ойн хязгаар 2048 MiB, хугацааны хязгаар 0.02 секунд байна.
- *circuit()* функцийг санах ойн хязгаарлалт 2048 MiB ба хугацааны хязгаарлалт нь 7 секунд байна.

Эцсийн үнэлгээний хувьд *alice()* болон *bob()* функцийг нэг тестийн тохиолдолд олон удаа дуудаж болно. Дуудлага бүрийн хувьд 0.02 секундын хязгаартай байна.

Дэд бодлогууд

А төрлийн дэд бодлого (12 оноо)

1,2,3-р дэд бодлого нь А төрлийнх бөгөөд энд $n = 1$ байна.

Эдгээр дэд бодлогууд нь дараах нэмэлт хязгаарлалттай:

- Дэд бодлого 1 (4 оноо): $m = 0$.
- Дэд бодлого 2 (4 оноо): $0 \leq m \leq 1$.
- Дэд бодлого 3 (4 оноо): $0 \leq m \leq 1000$.

В төрлийн дэд бодлого (54 оноо)

4,5,6-р дэд бодлого нь В төрлийнх бөгөөд энд:

- $0 \leq n \leq 30, \frac{n}{2} \leq m \leq n^2$.
- Нэг ижил илгээгч ба хүлээн авагчтай хоёр захидал байхгүй.
- Бүх гишүүдийн нэр Бобын оролтын хэсэгт байна (өөрөөр хэлбэл гишүүн бүр дор хаяж нэг захидал илгээсэн эсвэл дор хаяж нэг захидал хүлээн авсан)

Эдгээр дэд бодлого бүр дараах нэмэлт хязгаарлалтуудтай:

- Дэд бодлого 4 (24 оноо): $n = 26$, Бүх гишүүдийн нэр нь нэг жижиг үсгээр бичигдсэн байх ба Алисын оролтод a -аас z хүртэл дарааллаар байна.
- Дэд бодлого 5 (24 оноо): $n = 26$.
- Дэд бодлого 6 (6 оноо): Онцолсон хязгаарлалт байхгүй.

С төрлийн дэд бодлого (34 оноо)

7,8,9-р дэд бодлого нь С төрлийнх бөгөөд энд $0 \leq n \leq 700, 0 \leq m \leq 1000$.

Эдгээр дэд бодлогууд дараах нэмэлт хязгаарлалтуудтай:

- Дэд бодлого 7 (18 оноо): $n = 676$, Бүх гишүүдийн нэр нь хоёр жижиг үсгээс бүрдэх ба Алисын оролтод толь бичгийн дарааллаар байрлана (Тухайлбал $aa, ab, ac, \dots, az, ba, \dots, bz, ca, \dots, zz$).
- Дэд бодлого 8 (10 оноо): $n = 676$.
- Дэд бодлого 9 (6 оноо): Нэмэлт хязгаарлалт байхгүй.

Жишээ шалгагч (Grader)

Жишээ шалгагч оролтыг дараах форматаар уншина:

- Мөр 1: $n\ m$
- Мөр $2 + i$ ($0 \leq i \leq n - 1$): $names_i\ numbers_i$
- Мөр $2 + n + i$ ($0 \leq i \leq m - 1$): $senders_i\ recipients_i$.

Жишээ шалгагч гаралтыг дараах форматаар гаргана:

- Хэрэв программ нь амжилттай дуусвал жишээ шалгагч нь гишүүн бүрийн хувьд хэрэгжүүлсэн функцуудээр тооцоолсон эцсийн үр дүнгийн утгыг илэрхийлэх тус бүр нь бүхэл тоо агуулсан n мөрийг гаргах болно.
- Хэрэв программ нь амжилтгүй дуусвал жишээ шалгагч нь stdout руу юу ч гаргахгүй бөгөөд хавтас доторх *abc.log* файл руу алдааны мэдэгдлийг хэвлэнэ.
- Нэмж дурдахад жишээ шалгагч нь l_A, l_B, l утгууд болон функц бүрийн ажиллах хугацааг *abc.log* файл руу гаргана.

Жишээ шалгагч нь санах ойн хязгаар болон $n / m, l_A / l_B$ -ийн хувьд тэнцүү байх ёстой гэсэн хязгаарлалтыг шалгахгүй.