

Message

Айша менен Басма бири-бири менен кат жазышкан эки дос. Айшанын M атту каты бар, ал S биттеринин ырааттуулугу (б.а., нөл же бир), Бул катты Басмага жөнөткүсү келет. Айша Басма менен **пакеттерди** жөнөтүп байланышат. Пакет 0 дөн 30 га чейин индекстелген 31 биттин ырааттуулугу. Айша Басмага бир нече пакеттерди жөнөтүү менен M ди жөнөткүсү келет.

Тилекке каршы, Клеопатра Айша менен Басманын ортосундагы байланышты бузду жана пакеттерди **буза** алат. Башкача айтканда, ар бир пакетте Клеопатра биттердин так 15 индексин өзгөртүшү мүмкүн. Тактап айтканда, 31 узундуктагы C массиви бар, анда ар бир элемент же 0 же 1 болуп, төмөнкү мааниге ээ:

- $C[i] = 1$ i индексиндеги бит Клеопатра өзгөртө алышын көрсөтөт. Биз бул индекстерди **Клеопатра башкарган** деп атайбыз.
- $C[i] = 0$ i индексиндеги бит Клеопатра өзгөртө албашын көрсөтөт.

C массивинде так 15 бир жана 16 нөл бар. M ди жөнөтүп жатканда, Клеопатра башкарган индекстердин жайгашуусу бардык пакеттер үчүн бирдей бойдон калат. Айша Клеопатра кайсы 15 индекстерин башкарганын так билет. Басма 15 индекстер Клеопатра башкаруусун гана билип, ал кайсы индекстер экенин билбейт.

А Айша жөнөтүүнү чечкен пакет болсун (Биз аны **оригиналдуу пакет** деп атайбыз). Басма алган пакет B болсун (биз муну **өзгөртүлгөн пакет** деп атайбыз). Ар бир i үчүн $0 \leq i < 31$:

- эгерде Клеопатра i ($C[i] = 0$) индексиндеги битти башкарбаса, Басма Айша жөнөткөн i ($B[i] = A[i]$) битти алат,
- болбосо, эгерде Клеопатра i ($C[i] = 1$) индексиндеги битти башкарса, $B[i]$ маанисин Клеопатра чечет.

Ар бир пакетти жөнөткөндөн кийин дароо, Айша ар бир оригиналдуу пакетке тиешелүү болгон өзгөртүлгөн пакеттер кандай экенин үйрөнөт.

Айша бардык пакеттерди жөнөткөндөн кийин, Басма бардык өзгөртүлгөн пакеттерди **жөнөтүлгөн тартипте** алып, баштапкы катты кайра түзүш керек .

Сиздин милдетиңиз стратегияны иштеп чыгып ишке ашыруу. Айша M атту катты жөнөтүүсү жана Басма катты өзгөртүлгөн пакеттерден кайра түзүп аллуусу керек. Тактап айтканда, эки процедураны ишке ашыруу керек. Биринчи процедура Айшанын аракеттерин аткарат. Ага M атту кат жана C массиви берилет, жана Басмага катты өткөрүү үчүн бир нече пакеттерди

жөнөтүү керек. Экинчи процедура Басманын аракеттерин аткарат. Ага өзгөртүлгөн пакеттер берилет жана баштапкы катты кайра түзүп алуусу керек.

Implementation Details

Сиз ишке ашыруу керек болгон биринчи процедура:

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

- M : сүрөттөгөн S узундуктагы массив Айшанын Басмага жибергиси келген кабары.
- C : 31 узундуктагы массив Клеопатра тарабынан башкарган биттердин индекстерин көрсөтүү.
- Бул процедураны ар бир сыноо ишинде **2100дөн ашпаган жолу** чакырса болот.

Бул процедура пакетти жөнөтүү үчүн төмөнкү процедураны чакырышы керек:

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

- A : оригиналдуу пакет (узундугу 31 массив) Айша жиберген биттерди билдирет.
- Бул процедура өзгөртүлгөн B пакетин кайтарат Басма кабыл ала турган биттерди билдирет.
- Бул процедураны `send_message`тин ар бир чакыруусунда 100дөн ашпаган жолу чакырса болот.

Сиз ишке ашырууга тийиш болгон экинчи процедурасы:

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

- R : өзгөртүлгөн пакеттерди сүрөттөгөн массив. Пакеттер Айша ар бир `send_message` чалуусунда жөнөткөн пакеттерден келип чыгат жана Айша тарабынан **жөнөтүлгөн тартипте** берилет. R дин ар бир элементи өзгөртүлгөн пакетти билдирген 31 узундуктагы массив.
- Бул процедура S биттүү M ге барабар болгон массивди кайтарышы керек
- Бул процедураны ар бир сыноодо **бир нече жолу** чакырууга уруксаат этилет, бирок ар бир тиешелүү `send_message` чалуу үчүн **так бир жолу** чакырууга уруксаат этилет. **`receive_message` процедураны чакыруу тартиби** тиешелүү `send_message` чалууларынын тартиби менен сөзсүз эле бирдей эмес.

Баалоо системасында `send_message` жана `receive_message` процедуралары **өзүнчө программаларда** чакырылаарын эске алыңыз.

Constraints

- $1 \leq S \leq 1024$
- C да так 31 элементтер бар, анын ичинен 16сы 0 ге жана 15и 1 ге барабар.

Subtasks and Scoring

Эгерде кандайдыр бир сыноо учурларында, `send_packet` процедурасына чалуулар жогоруда айтылган эрежелерге ылайык келбесе, же `receive_message` процедурасына чалуулардын биринин кайтаруу мааниси туура эмес, ал сыноо иши үчүн чечимиңиздин упайы 0 болот.

Болбосо, Q бардык сыноо учурлары боюнча `send_message` чакырууларынын арасында `send_packet` процедурасына чалуулардын максималдуу саны болсун.

Ошондой эле X :

- 1ге барабар болгон учур бул, эгерде $Q \leq 66$ болсо
- 0.95^{Q-66} ге барабар болгон учур бул, эгерде $66 < Q \leq 100$

Андан кийин, упай төмөнкүдөй эсептелет:

Subtask	Упай	Кошумча чектөөлөр
1	$10 \cdot X$	$S \leq 64$
2	$90 \cdot X$	Эч кандай кошумча чектөөлөр.

Грейдер **адаптивдүү** болушу мүмкүн экенин белгилей кетүү керек. Бул `send_packet` функциясы кайтарган маанилер анын киргизүү аргументтеринен эле эмес, ошондой эле көптөгөн башка нерселерден, анын ичинде бул процедурага мурунку чалуулардагы киргизүүлөр жана кайтаруу маанилеринен жана грейдер тарабынан түзүлгөн псевдо-рандом сандардан көз каранды болушу мүмкүн дегенди билдирет. Грейдер **детерминистикалык** деген мааниде, эгерде сиз аны эки жолу иштетсеңиз жана экөөндө тең бир эле пакеттерди жөнөтсөңүз, ал аларга бирдей өзгөртүүлөрдү киргизет.

Example

Төмөнкү чакырыкты карап көрөлү.

```
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])
```

Айша Басмага жөнөтүүгө аракет кылган кат $[0, 1, 1, 0]$. 0 дөн 15 ке чейинки индекстеги биттерди Клеопатра өзгөртө албайт, ал эми 16 дан 30 га чейинки индекстеги биттерди Клеопатра өзгөртө алат.

Бул мисал үчүн, келгиле, Клеопатранын жүрүм-туруму детерминисттик деп ойлойлу жана ал өзү башкарган ырааттуу биттерди 0 жана 1 алмаштырат, башкача айтканда, ал башкарган биринчи индекске 0 дайындайт (биздин учурда 16 индекси) , ал башкарган экинчи индекске 1 (индекс 17) , ал башкарган үчүнчү индекске 0 (индекс 18) ушундай ирээтте улантат.

Айша баштапкы билдирүүдөн эки битти бир пакетте төмөнкүчө жөнөтүүнү чече алат: биринчи битти өзү башкарган биринчи 8 индексине, экинчи битти өзү башкарган кийинки 8 индексине жөнөтөт.

Андан кийин Айша төмөнкү пакетти жөнөтүүнү чечет:

```
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 акыркы 15 индекстери менен биттерди өзгөртө аларын эске алыңыз, ошондуктан Айша аларды каалаганча орното алат, анткени алар өзгөрүшү мүмкүн. Клеопатранын болжолдонгон стратегиясы менен процедура төмөнкүдөй кайтарылат: $[0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0]$.

Айша экинчи пакетте M акыркы эки бит жөнөтүүнү мурункудай эле чечет:

```
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])
```

Клеопатранын болжолдонгон стратегиясы менен процедура кайтып келет: $[1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0]$.

Айша дагы пакеттерди жөнөтө алат, бирок ал жөнөтпөйт.

Андан кийин грейдер төмөнкү процедураны чакырат:

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

Васма M билдирүүсүн төмөнкүдөй калыбына келтирет. Ар бир пакеттен ал эки жолу катары менен болгон биринчи эки битти жана эки жолу катары менен болгон акыркы эки битти алат. Башкача айтканда, биринчи пакеттен $[0, 1]$ биттерин, ал эми экинчи пакеттен $[1, 0]$

биттерин алат. Аларды бириктирүү менен ал $[0, 1, 1, 0]$ билдирүүсүн калыбына келтирет, бул `receive_message` чалуу үчүн туура кайтаруу мааниси.

Клеопатранын болжолдонгон стратегиясы менен жана 4 узундуктагы билдирүүлөр үчүн Басманын мындай мамилеси C баасына карабастан M туура калыбына келтирээрин көрсөтсө болот. Бирок, бул жалпы учурда туура эмес.

Sample Grader

Үлгү грейдер адаптацияланган эмес. Анын ордуна, Клеопатра жогорудагы мисалда сүрөттөлгөндөй, ал башкарган ырааттуу биттерди 0 жана 1 биттери менен толтурат.

Киргизүү форматы: **Киргизүүнүн биринчи сабында сценарийлердин санын көрсөтүүчү бүтүн T бар.** T сценарийлери андан кийин болот. Алардын ар бири төмөнкү форматта берилет:

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

Чыгуу форматы: Үлгү грейдер T сценарийлеринин ар биринин жыйынтыгын төмөнкү форматта киргизүүдө берилген тартипте жазат:

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

Бул жерде, K `send_packet` га чалуулардын саны, D – `receive_message` тарабынан кайтарылган билдирүү жана L анын узундугу.