

# Message

Айша мен Басма бір-бірімен хат алмасатын екі дос. Айшада Басмаға жібергісі келетін S биттерінің тізбегі (яғни, нөлдер немесе бірлер) болатын M хабарламасы бар. Айша Басмаға **пакеттер** жіберу арқылы хабарласады. Пакет – 0-ден 30-ға дейінгі индекстермен нөмірленген 31 биттерінің тізбегі. Айша Басмаға бірнеше пакеттер жіберу арқылы M хабарламасын жібергісі келеді.

Өкінішке орай, Клеопатра Айша мен Басма арасындағы байланысты бұзып пакеттерді өзгертуге қабілетті. Яғни, әрбір пакетте Клеопатра дәл 15 индекстегі биттерді өзгерте алады. Атап айтқанда, ұзындығы 31 болатын C массиві бар, онда келесі жолмен әрбір элемент 0 немесе 1 болады:

- C[i]=1 Клеопатра i-ші индекстегі битті өзгерте алатынын көрсетеді. Біз бұл индекстерді Клеопатра **басқаратын** деп атаймыз.
- C[i] = 0 Клеопатра i-ші индекстегі битті өзгерте алмайтынын көрсетеді.

C массивінде дәл 15 бір және 16 нөл бар. M хабарламасын жіберу кезінде Клеопатра басқаратын индекстер жинағы барлық пакеттер үшін бірдей болып қалады. Айша Клеопатраның қандай 15 индекстерін басқаратынын нақты біледі. Басма тек Клеопатраның 15 индекстерді басқаратынын біледі, бірақ ол қандай индекстер екенін білмейді.

Айша жіберетін пакет A болсын (біз оны **түпнұсқа пакет** деп атаймыз). Басма қабылдайтын пакет B болсын (біз оны **өзгертілген пакет** деп атаймыз).  $0 \le i < 31$  болатындай әрбір i үшін:

- ullet егер Клеопатра i-ші индекстегі битті басқармаса (C[i]=0), Басма Айша жіберген i-ші битін алады (B[i]=A[i]),
- ullet әйтпесе, егер Клеопатра i-ші индекстегі битті басқарса (C[i]=1), B[i] мәнін Клеопатра шешеді.

Әр пакетті жібергеннен кейін Айша бірден сәйкес өзгертілген пакетті біледі.

Айша барлық пакеттерді жіберген соң, Басма барлық өзгертілген пакеттерді **олар жіберілген ретімен** алады және бастапқы M хабарламасын қайта құруы керек.

Сіздің міндетіңіз Айшаның Басмаға M хабарламасын жіберуге мүмкіндік беретін стратегияны ойлап табу және жүзеге асыру, осылайша Басма өзгертілген пакеттерден M хабарламасын қалпына келтіруі керек. Атап айтқанда, сіз екі функцияны орындауыңыз керек. Бірінші

функция Айшаның әрекеттерін орындайды. Оған M хабарламасы және C массиві беріледі және хабарламаны Басмаға тасымалдау үшін бірнеше пакеттер жіберуі керек. Екінші функция Басманың әрекеттерін орындайды. Оған өзгертілген пакеттер беріледі және M бастапқы хабарламасын қалпына келтіруі керек.

#### **Implementation Details**

Сізге іске асыруға керек бірінші функция:

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

- M: Айша Басмаға жібергісі келетін хабарламаны сипаттайтын ұзындығы S болатын массив.
- ullet C: Клеопатра басқаратын биттердің индекстерін көрсететін ұзындығы 31 болатын массив.
- Бұл функция әрбір сынақ жағдайында ең көбі 2100 рет шақырылуы мүмкін.

Бұл функция пакетті жіберу үшін келесі функцияны шақыруы керек:

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

- A: Айша жіберген биттерді білдіретін түпнұсқа пакет (ұзындығы 31 болатын массив).
- ullet Бұл функция Басма қабылдайтын биттерді білдіретін B өзгертілген пакетін қайтарады.
- $\bullet$  Бұл функцияны send\_message функциясының әр шақыруында ең көбі 100 рет шақыруға болады.

Сізге іске асыруға керек екінші функция:

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

- R: өзгертілген пакеттерді сипаттайтын массив. Пакеттер Айшаның бір send\_message шақыруында жіберген пакеттерінен туындайды және **оларды Айша жіберген ретпен** береді. R массивінің әрбір элементі өзгертілген пакетті білдіретін ұзындығы 31 болатын массив болып табылады.
- ullet Бұл функция M бастапқы хабарламасына тең S биттерден тұратын массив қайтаруы керек.
- Бұл функция әрбір сынақ жағдайында **бірнеше рет** шақырылуы мүмкін. Әрбір сәйкес send\_message шақырылуы үшін **дәл бір рет** шақырылады. receive\_message шақыруларының реті send\_message шақыруларының ретімен міндетті түрде бірдей емес.

Бағалау жүйесінде send\_message және receive\_message функциялары **екі бөлек бағдарламада** шақырылатынын ескеріңіз.

#### **Constraints**

- 1 < *S* < 1024
- C дәл 31 элементтерден тұрады, оның ішінде 16 элементтер 0-ге және 15 элементтер 1 -ге тең.

#### Subtasks and Scoring

Тесттердің кез келгенінде send\_packet функциясына шақырулар жоғарыда аталған ережелерге сәйкес келмесе немесе receive\_message функциясына шақырулардың кез келгенінің қайтару мәні қате болса, сіздің шешіміңіздің ұпайы бұл тестте 0 болады.

Әйтпесе, барлық тесттердің send\_message шақыруларының ішінде send\_packet функциясын ең көп шақырулар саны Q болсын. Сонымен X мынаған тең болсын:

- ullet 1 , егер  $Q \leq 66$  болса
- ullet  $0.95^{Q-66}$  , егер  $66 < Q \leq 100$  болса

Содан кейін ұпай келесі түрде есептеледі:

Ішкі есеп	Ұпай	Қосымша шектеулер
1	$10 \cdot X$	$S \leq 64$
2	$90 \cdot X$	Қосымша шектеулер жоқ.

Кейбір жағдайларда грейдер **бейімделгіш** болуы мүмкін екенін ескеріңіз. Бұл send\_packet қайтаратын мәндер оның енгізу аргументтеріне ғана емес, сонымен қатар осы функцияның алдыңғы шақыруларының кірістері мен қайтару мәндерін және грейдер жасаған жалған кездейсоқ сандарды қоса алғанда, көптеген басқа нәрселерге байланысты болуы мүмкін дегенді білдіреді. Грейдер **детерминистикалық** болып табылады. Егер сіз оны екі рет іске қоссаңыз және екеуінде де бірдей пакеттерді жіберсеңіз, ол оларға бірдей өзгерістер енгізеді.

### Example

Келесі шақыруды қарастырыңыз.

Айша Басмаға жібермекші болған хабарлама [0,1,1,0]. Индекстері 0-ден 15-ке дейінгі биттерді Клеопатра өзгерте алмайды, ал индекстері 16-дан 30-ға дейінгі биттерді Клеопатра

өзгерте алады.

Осы мысал үшін, Клеопатра өзі басқаратын биттерді кезектесіп 0-ге және 1-ге ауыстырады деп есептейік, яғни ол өзі басқаратын бірінші индекске 0 қояды (біздің жағдайда 16-ші индекс), екінші басқаратын индекске 1 қояды (17-ші индекс), ол басқаратын үшінші индекске 0 (18-ші индекс) және т.б.

Айша бастапқы хабарламадан екі битті бір пакетте келесідей жіберуге шешім қабылдай алады: ол бірінші битті өзі басқаратын бірінші 8 индекстерімен және екінші битті өзі басқаратын келесі 8 индекстерімен жібереді.

Содан кейін Айша келесі пакетті жіберуді таңдайды:

```
send_packet([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])
```

Айша екінші пакетте M-ның соңғы екі битін бұрынғыдай жіберуге шешім қабылдады:

Клеопатраның бұл стратегиясымен функция келесіні қайтарады: [1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,1,0,1,0,1,0,1,0,1,0,1,0] .

Айша көбірек пакеттер жібере алады, бірақ ол жібермеуді шешті.

Содан кейін грейдер келесі функцияны шақырады:

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

Басма M хабарламасын келесідей қалпына келтіреді. Әр пакеттен ол қатарынан екі рет кездесетін бірінші битті және қатарынан екі рет кездесетін соңғы битті алады. Яғни, бірінші пакеттен ол [0,1] биттерін, ал екінші пакеттен [1,0] биттерін алады. Оларды біріктіру арқылы ол [0,1,1,0] хабарламасын қалпына келтіреді, бұл receive\_message шақыруының дұрыс қайтару мәні.

Клеопатраның бұл стратегиясымен және 4 ұзындықтағы хабарламалар үшін Басманың бұл стратегиясы C мәніне қарамастан M хабарламасын дұрыс қалпына келтіретінін көрсетуге болады. Алайда, жалпы жағдайда бұл дұрыс емес.

## Sample Grader

Үлгі грейдер бейімделмейді. Оның орнына, Клеопатра жоғарыдағы мысалда сипатталғандай, өзі басқаратын биттерді кезектесіп 0 және 1 биттерімен толтырады.

Енгізу форматы: **Енгізудің бірінші жолында тест жағдайлар санын көрсететін** T **бүтін саны бар.** Одан кейін T тест жағдайлары сипатталған. Олардың әрқайсысы келесі форматта берілген:

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

Шығару форматы: Үлгі грейдер T тест жағдайларының әрқайсысының нәтижесін келесі форматта берілген ретпен жазады:

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

Мұнда K - send\_packet функциясының шақырулар саны, D - receive\_message функциясы қайтарған хабарлама және L - оның ұзындығы.