

## Žinutė

Aiša ir Basma yra dvi draugės, kurios susirašinėja tarpusavyje. Aiša parašė žinutę  $M$ , kurią sudaro  $S$  bitų (t. y. nulių arba vienetų) seka, ir norėtų pasiųsti ją Basmai. Aiša komunikuoja su Basma siųsdama jai **paketus**. Paketas yra 31 bito seka. Bitai sunumeruoti nuo 0 iki 30. Aiša norėtų pasidalinti su Basma žinute  $M$ , pasiųsdama jai kažkiek paketų.

Deja, Kleopatra įsiterpė į komunikacijos kanalą tarp Aišos ir Basmos ir dabar gali **suteršti** paketus. Tai reiškia, kad kiekviename pakete Kleopatra gali keisti bitus lygiai 15-oje pozicijų. Kitaip tariant, egzistuoja toks masyvas  $C$ , kurio ilgis 31 ir kurio kiekvienas elementas lygus 0 arba 1, bei:

- $C[i] = 1$  reiškia, kad Kleopatra gali keisti  $i$ -ąjį bitą. Tokiu atveju sakysime, kad  $i$ -ąją poziciją Kleopatra **valdo**.
- $C[i] = 0$  reiškia, kad Kleopatra negali pakeisti  $i$ -ojo bito.

$C$  masyve yra lygiai 15 vienetų ir 16 nulių. Kol siunčiama žinutė  $M$ , Kleopatos valdomos pozicijos išlieka tos pačios visiems paketams. Aiša tiksliai žino, kurias 15 pozicijų valdo Kleopatra. Basma žino tik tai, kad Kleopatra valdo 15 pozicijų, bet nežino, kurias konkrečiai pozicijas ji valdo.

Tarkime, kad Aiša nutaria pasiųsti paketą  $A$  (kurį mes pavadinsime **pradiniu paketu**), o Basma gauna paketą  $B$  (kurį vadinsime **užterštu paketu**). Kiekvienam  $i$ , kuriam  $0 \leq i < 31$ :

- Jei Kleopatra nevaldo  $i$ -osios pozicijos ( $C[i] = 0$ ), Basmai atkeliauja tokia  $i$ -ojo bito reikšmė, kokį siuntė Aiša ( $B[i] = A[i]$ ),
- kitu atveju, jei Kleopatra valdo  $i$ -ąją poziciją ( $C[i] = 1$ ),  $B[i]$  reikšmę nusprendžia Kleopatra.

Iš karto po kiekvieno paketo išsiuntimo Aiša sužino atitinkamo užteršto paketo reikšmes.

Po to, kai Aiša pasiunčia visus paketus, Basma gauna visus užterštus paketus **ta pačia tvarka, kuria jie buvo siųsti**, ir turi atkurti pradinę žinutę  $M$ .

Sugalvokite strategiją, kaip Aiša galėtų pasiųsti žinutę  $M$  Basmai taip, kad Basma galėtų atkurti žinutę  $M$  iš užterštų paketų. Jums reikia parašyti dvi procedūras. Pirmoji procedūra atlieka Aišos veiksmus. Ši procedūra gauna žinutę  $M$  bei masyvą  $C$ , ir turi išsiųsti kažkiek paketų Basmai. Antroji procedūra atlieka Basmos veiksmus. Ši procedūra gauna užterštus paketus ir turi grąžinti pradinę žinutę  $M$ .

# Realizacija

Pirmoji procedūra, kurią jums reikia parašyti, yra:

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

- $M$ :  $S$  ilgio masyvas, sudarantis žinutę, kurią Aiša nori pasiųsti Basmai.
- $C$ : 31 elemento ilgio masyvas, nurodantis Kleopatros valdomas pozicijas.
- Ši procedūra gali būti iškviesta **daugiausiai 2100 kartų** kiekvienam testui.

Ši procedūra turėtų kviesti žemiau esančią procedūrą, kad pasiųstų paketą:

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

- $A$ : pradinis paketas (31-o elemento ilgio masyvas), nurodantis Aišos pasiųstų bitų reikšmes.
- Ši procedūra grąžina užterštą paketą  $B$ , tai yra bitus, kuriuos gaus Basma.
- Ši procedūra gali būti iškviesta daugiausiai 100 kartų kiekvienam `send_message` iškvietai.

Antroji procedūra, kurią jums reikia parašyti, yra:

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

- $R$ : masyvas, nurodantis užterštus paketus. Paketai gaunami iš Aišos išsiųstų paketų viename `send_message` iškvietai ir pateikiami **ta tvarka, kuria juos išsiuntė** Aiša. Kiekvienas  $R$  elementas yra masyvas, kurio ilgis 31, reiškiantis užterštą paketą.
- Ši procedūra turi grąžinti  $S$  bitų masyvą, lygų pradinei žinutei  $M$ .
- Ši procedūra gali būti iškviesta **kelis kartus** kiekvienam testui, **lygiai vieną kartą** kiekvienam `send_message` iškvietai. `receive_message` **procedūros iškvietai tvarka** nebūtinai tokia pati, kaip `send_message` iškvietai tvarka.

Atkreipkite dėmesį, kad `send_message` ir `receive_message` vertinimo sistemoje iškviečiami **dviejose skirtingose programose**.

## Ribojimai

- $1 \leq S \leq 1024$
- $C$  turi lygiai 31 narį, iš kurių 16 yra lygūs 0 ir 15 yra lygūs 1.

## Dalinės užduotys ir vertinimas

Jeigu bet kuriame teste procedūros `send_packet` iškvietai neatitinka aukščiau aprašytų reikalavimų, arba jeigu bet kurio `receive_message` iškvietai grąžinta reikšmė neteisinga, už tą testą skiriama 0 taškų.

Kitu atveju tegul  $Q$  žymi didžiausią procedūros `send_packet` iškvietimų skaičių tarp visų `send_message` iškvietimų tarp visų testų. Taip pat tegul  $X$  būna lygus:

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

Tada taškai skaičiuojami taip:

Dalinė užduotis	Taškai	Ribojimai
1	$10 \cdot X$	$S \leq 64$
2	$90 \cdot X$	Jokių papildomų ribojimų.

Atkreipkite dėmesį, kad kai kuriais atvejais vertinimo programa gali būti **adaptyvi**. Tai reiškia, kad `send_packet` grąžintos vertės gali priklausyti ne tik nuo jai duotų argumentų, bet ir nuo daugelio kitų dalykų, įskaitant ankstesnių procedūros iškvietimų argumentų bei grąžintų verčių reikšmių, bei vertinimo programos sugeneruotų pseudoatsitiktinių skaičių. Vertinimo programa deterministinė: jei ji paleidžiama du kartus ir abu kartus siunčiami tie patys paketai, paketams bus atlikti tie patys pakeitimai.

## Pavyzdys

Panagrinėkime tokį iškvietimą.

```
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, 1])
```

Aiša bando Basmai nusiųsti žinutę `[0, 1, 1, 0]`. Kleopatra negali pakeisti bitų, esančių pozicijose nuo 0 iki 15, tačiau gali keisti bitus, esančius pozicijose nuo 16 iki 30.

Šiame pavyzdyje laikykime, kad Kleopatra valdomas pozicijas pakaitomis užpildo 0-iais ir 1-ais, t. y. ji priskiria 0 pirmajai pozicijai, kurią ji valdo (šiuo atveju pozicijai 16), 1 -- antrajai pozicijai, kurią ji valdo (pozicijai 17), 0 -- trečiajai pozicijai, kurią ji valdo (pozicijai 18), ir t.t.

Aiša gali nuspręsti nusiųsti du bitus pradinės žinutės viename pakete tokiu būdu: ji nusiųs pirmąjį bitą pirmose 8 pozicijose, kurias ji valdo, o antrąjį bitą -- kitose 8 pozicijose, kurias ji valdo.

Aiša tuomet siųs tokį paketą:

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

Atkreipkite dėmesį, kad Kleopatra gali pakeisti bitus, paskutinėse 15 pozicijų, taigi šių bitų reikšmės Aiša gali laisvai pasirinkti, nes jos gali būti pakeistos.

Turint omeny aprašytą Kleopatros strategiją, procedūra grąžina:

[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].

Aiša nusprendžia antrame pakete nusiųsti du paskutinius žinutės  $M$  tuo pat būdu kaip anksčiau:

```
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])
```

Pritaikius Kleopatros strategiją procedūra grąžina:

[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].

Aiša gali siųsti daugiau paketų, bet pasirenka to nedaryti.

Vertinimo programa iškviečia procedūrą tokiu būdu:

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

Basma taip atstato žinutę  $M$ . Iš kiekvieno paketo ji paima pirmą bitą, kuris pasikartoja du kartus iš eilės bei paskutinįjį bitą, kuris pasikartoja du kartus iš eilės. Tai yra, iš pirmojo paketo ji paima bitus [0, 1], o iš antrojo -- [1, 0]. Juos sudeda kartu ir gauna žinutę [0, 1, 1, 0], kuri yra teisingas sprendinys ir turėtų būti grąžinama reikšmė šiam receive\_message iškvietimui.

Galima parodyti, kad šis Basmos būdas atkurti žinutę  $M$  visada veiks, jei Kleopatra naudos aukščiau aprašytą strategiją, kai žinutės ilgis yra 4, nepriklausomai nuo  $C$  reikšmės. Deja, tai nėra teisinga strategija bendru atveju.

## Pavyzdinė vertinimo programa

Pvayzdinė vertinimo programa nėra adaptyvi. Vietoj to, Kleopatra užpildo iš eilės einančias valdomas pozicijas pakaitomis 0-iais ir 1-ais, kaip apibūdinta pavyzdyje aukščiau.

Įvesties formatas: **Pirmoje eilutėje yra sveikasis skaičius  $T$ , nurodantis scenarijų kiekį.** Toliau aprašomi  $T$  scenarijų. Kiekvienas iš jų apibūdinamas šiuo formatu:

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

Išvesties formatas: Pavyzdinė vertinimo programa išveda kiekvieno iš  $T$  scenarijų rezultatą ta pačia tvarka, kokia buvo pateikta įvestis, tokiu formatu:

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

Čia  $K$  yra `send_packet` iškvietyimų kiekis,  $D$  yra `receive_message` grąžinta žinutė o  $L$  yra jos ilgis.