Sporočilo

Aisha in Basma sta prijateljici, ki si dopisujeta. Aisha ima sporočilo M, ki je zaporedje S bitov (tj. ničel ali enic), ki ga želi poslati Basmi. Aisha komunicira z Basmo tako, da ji pošilja **pakete**. Paket je zaporedje 31 bitov, oštevilčenih od 0 do 30. Aisha želi poslati sporočilo M Basmi tako, da ji pošlje nekaj paketov.

Na žalost je Kleopatra kompromitirala komunikacijo med Aisho in Basmo in lahko **spremeni** pakete. To pomeni, da lahko Kleopatra v vsakem paketu spremeni bite na točno 15 indeksih. Natančneje, obstaja polje C dolžine 31, kjer je vsak element bodisi 0 bodisi 1, z naslednjim pomenom:

- $C[i] = 1$ pomeni, da Kleopatra lahko spremeni bit z indeksom i. Te indekse imenujemo **nadzorovani** s strani Kleopatre.
- $C[i] = 0$ pomeni, da Kleopatra ne more spremeniti bita z indeksom i.

Polje C vsebuje natanko 15 enic in 16 ničel. Med pošiljanjem sporočila M ostaja nabor indeksov, ki jih nadzoruje Kleopatra, in je enak za vse pakete. Aisha natančno ve, katerih 15 indeksov nadzoruje Kleopatra. Basmi je znano le, da Kleopatra nadzoruje 15 indeksov, vendar ne ve katerih.

Naj bo A paket, ki se ga Aisha odloči poslati (imenujmo ga **izvirni paket**). Naj bo B paket, ki ga prejme Basma (imenujemo ga **spremenjeni paket**). Za vsak i, kjer $0 \leq i < 31$ velja:

- če Kleopatra ne nadzoruje bita z indeksom i ($C[i] = 0$), Basma prejme bit i tako, kot ga je poslala Aisha ($B[i] = A[i]$),
- sicer, če Kleopatra nadzoruje bit z indeksom i ($C[i] = 1$), vrednost $B[i]$ določi Kleopatra.

Takoj po pošiljanju vsakega paketa Aisha izve, kakšen je spremenjeni paket.

Ko Aisha pošlje vse pakete, Basma prejme vse spremenjene pakete **v vrstnem redu, kot so bili poslani** in mora rekonstruirati izvirno sporočilo M.

Vaša naloga je zasnovati in implementirati strategijo, ki bi omogočila Aishi, da pošlje sporočilo M Basmi, tako da Basma lahko obnovi M iz spremenjenih paketov. Natančneje, implementirati morate eno proceduro in eno funkcijo. Prva procedura izvaja dejanja Aishe. Sprejme sporočilo M ter polje C, in mora poslati nekaj paketov, za prenos sporočila Basmi. Druga funkcija izvaja dejanja Basme. Sprejme spremenjene pakete in mora obnoviti izvirno sporočilo M.

Podrobnosti implementacije

Procedura, ki jo morate implementirati, je:

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

- M: polje dolžine S, ki opisuje sporočilo, ki ga želi Aisha poslati Basmi.
- C: polje dolžine 31, ki označuje indekse bitov, ki jih nadzoruje Kleopatra.
- Proceduro se lahko kliče **največ 2100-krat** pri vsakem testnem primeru.

Ta procedura mora poklicati naslednjo funkcijo za pošiljanje paketa:

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

- A: izvirni paket (polje dolžine 31), ki predstavlja bite, ki jih pošlje Aisha.
- Funkcija vrne spremenjeni paket B, ki predstavlja bite, ki jih bo prejela Basma.
- Funkcijo se lahko kliče največ 100-krat v vsakem klicu `send_message`.

Funkcija, ki jo morate implementirati, je:

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

- R: polje, ki opisuje spremenjene pakete. Paketi izvirajo iz paketov, ki jih je Aisha poslala z enim klicem `send_message` in so podani **v vrstnem redu, kot so bili poslani** s strani Aishe. Vsak element R je polje dolžine 31, ki predstavlja spremenjeni paket.
- Funkcija mora vrniti polje dolžine S bitov, ki je enako izvirnemu sporočilu M.
- Funkcijo se lahko pokliče **večkrat** pri vsakem testnem primeru, **natanko enkrat** za vsak ustrezni klic `send_message`. **Vrstni red klicev funkcije** `receive_message` ni nujno enak vrstnemu redu kakor so bili klici `send_message`.

Upoštevajte, da se v ocenjevalnem sistemu procedura `send_message` in funkcija `receive_message` kličeta v **dveh ločeni programih**.

Omejitve

- $1 \leq S \leq 1024$
- C ima natanko 31 elementov, od katerih je 16 enakih 0 in 15 enakih 1.

Podnaloge in točkovanje

Če v katerem koli testnem primeru klici postopka `send_packet` ne ustrezajo zgoraj omenjenim pravilom, ali če je povratna vrednost katerega koli klica funkcije `receive_message` napačna, bo ocena vaše rešitve za ta testni primer 0.

V nasprotnem primeru, naj bo Q največje število klicev procedure `send_packet` med vsemi klici `send_message` vseh testnih primerih. Naj bo X enako:

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

Točkuje se na naslednji način:

Podnaloga	Točke	Dodatne omejitve
1	$10 \cdot X$	$S \leq 64$
2	$90 \cdot X$	Brez dodatnih omejitev.

Upoštevajte, da je v nekaterih primerih obnašanje ocenjevalnika lahko **prilagodljivo**. To pomeni, da se vrednosti, ki jih vrne `send_packet`, lahko razlikujejo ne samo glede na vhodne argumente, temveč tudi glede na mnoge druge stvari, vključno z vhodi in povratnimi vrednostmi prejšnjih klicev te procedure, kot tudi psevdonaključnimi števili, ki jih generira ocenjevalnik. Ocenjevalnik je **determinističen** v smislu, da če ga zaženete dvakrat in v obeh primerih pošljete enake pakete, bo na njih izvedel enake spremembe.

Primer

Razmislite o naslednjem klicu.

```
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])
```

Sporočilo, ki ga Aisha poskuša poslati Basmi, je $[0, 1, 1, 0]$. Bitov, z indeksi od 0 do 15, Kleopatra ne more spremeniti, medtem ko bite, z indeksi od 16 do 30, Kleopatra lahko spremeni.

Za namene tega primera predpostavimo, da Kleopatra zapolni zaporedne bite, ki jih nadzoruje, izmenično z 0 in 1, tj. dodeli 0 prvemu indeksu, ki ga nadzoruje (v našem primeru indeks 16), 1 drugemu indeksu, ki ga nadzoruje (indeks 17), 0 tretjemu indeksu, ki ga nadzoruje (indeks 18), in tako naprej.

Aisha se lahko odloči poslati dva bita iz izvirnega sporočila, v enem paketu na naslednji način: prvi bit bo poslala preko prvih 8 indeksov, ki jih nadzoruje, in drugi bit preko naslednjih 8 indeksov, ki jih nadzoruje.

Aisha se nato odloči poslati naslednji paket:

```
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])
```

Upoštevajte, da lahko Kleopatra spremeni bite na zadnjih 15 indeksih, zato jih lahko Aisha nastavi poljubno, saj bodo morda spremenjeni. S predpostavljeno Kleopatrino strategijo, funkcija vrne: $[0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1]$.

Aisha se odloči poslati zadnja dva bita M v drugem paketu, na podoben način kot prej:

```
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, 0])
```

S predpostavljeno Kleopatrino strategijo, funkcija vrne: $[1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0]$.

Aisha lahko pošlje več paketov, vendar se odloči, da jih ne bo.

Ocenjevalnik nato izvede naslednji klic funkcije:

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

Basma obnovi sporočilo M na naslednji način. Iz vsakega paketa vzame prvi bit, ki se pojavi dvakrat zapored, ter zadnji bit, ki se pojavi dvakrat zapored. Tako iz prvega paketa vzame bite $[0,1]$, in iz drugega paketa vzame bite $[1,0]$. S tem ko jih združi, obnovi sporočilo $[0,1,1,0]$, kar je pravilna povratna vrednost za ta klic `receive_message`.

Lahko se pokaže, da s predpostavljeno strategijo Kleopatre in za sporočila dolžine 4, ta Basmin pristop pravilno obnovi M, ne glede na vrednost C. Vendar, v splošnem primeru ni pravilen.

Vzorčni ocenjevalnik

Vzorčni ocenjevalnik ni prilagodljiv. Kleopatra zapolni zaporedne bite, ki jih nadzoruje, izmenično z biti 0 in 1, kakor je opisano v zgornjem primeru.

Oblika vhoda: **Prva vrstica vhoda vsebuje celo število T, ki določa število scenarijev.** Sledi T scenarijev. Vsak od njih je podan v naslednji obliki:

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

Oblika izhoda: Vzorčni ocenjevalnik izpiše rezultat vsakega izmed T scenarijev v enakem vrstnem redu kakor so podani v vhodu, v naslednji obliki:

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

Tukaj je K število klicev `send_packet`, D je sporočilo, ki ga vrne `receive_message` in L je njegova dolžina.