

Skilaboð

Kristinn og Þórhallur eru tveir vinir sem skrifa skilaboð til hvors annars. Kristinn er með skilaboð M sem er runa af S bitum, þar sem hver biti er núll eða ás, sem hann vill senda til Þórhalls. Kristinn á samskipti við Þórhall með því að senda honum **pakka**. Pakki er runa af 31 bitum með vísa frá 0 til 30. Kristinn vill senda skilaboðið M til Þórhalls með því að senda honum einhvern fjölda pakka.

Því miður er Eva að setja samskiptin í hættu og getur **spillt** pökkunum. Það þýðir að Eva getur breytt bitum í sérhverjum pakka á nákvæmlega 15 vísur. Þá sérstaklega, það er til fylki C sem er að lengd 31, þar sem sérhvert stak er annað hvort 0 eða 1 með eftirfarandi merkingu:

- $C[i] = 1$ þýðir að Eva geti breytt bitanum með vísi i . Við segjum að þessir bitar séu undir **stjórn** Evu.
- $C[i] = 0$ þýðir að Eva geti ekki breytt bitanum með vísi i .

Fylkið C inniheldur nákvæmlega 15 ása og 16 núll. Þegar skilaboð M er sent, þá er mengi vísa sem eru undir stjórn Evu óbreytt fyrir alla pakka. Kristinn veit nákvæmlega hvaða 15 vísar eru undir stjórn Evu. Þórhallur veit einungis að 15 vísar eru undir stjórn Evu, en hann veit ekki hvaða vísar það eru.

Segjum að A sé pakki sem Kristinn ákveður að senda, sem við köllum **upprunalega pakkann**. Segjum að B sé pakkinn sem Þórhallur tekur á móti, sem við köllum **spillta pakkann**. Fyrir sérhvert i , þar sem $0 \leq i < 31$:

- ef Eva stjórnar ekki bitanum með vísi i , eða $C[i] = 0$, þá fær Þórhallur bitann i eins og Kristinn sendi hann, eða $B[i] = A[i]$.
- annars, ef Eva stjórnar bitanum með vísi i , eða $C[i] = 1$ þá ákvarðar Eva gildið á $B[i]$.

Strax eftir að senda hvern pakka fær Kristinn að vita hver samsvarandi spillti pakki er.

Eftir að Kristinn sendir alla pakkana fær Þórhallur spilltu pakkana í sömu röð og þeir voru sendir og þarf hann að endurbyggja upprunalega skilaboðið M .

Verkefni þitt er að útbúa og útfæra aðferð sem leyfir Kristni að senda skilaboðið M til Þórhalls, þannig að Þórhallur getur endurheimt M frá spilltu pökkunum. Þá sérstaklega skaltu útfæra tvær stefjur. Fyrri stefjan framkvæmir aðgerðir Kristins. Hún fær skilaboð M og fylkið C , og á að senda einhverja pakka sem senda skilaboðið til Þórhalls. Seinni stefjan framkvæmir aðgerðir Þórhalls. Hún fær spilltu pakkana og á að endurheimta upprunalega skilaboðið M .

Úfærslusmáatriði

Fyrri stefjan sem þú skalt útfæra er:

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

- M : fylki sem hefur lengd N og lýsir skilaboðinu sem Kristinn vill senda Þórhalli.
- C : fylki sem hefur lengd 31 og segir til um hvaða vísar eru undir stjórn Evu.
- Leyfilegt er að kalla í þessa stefju **mest 2100 sinnum**.

Þessi stefja skal kalla í eftirfarandi stefju til að senda pakka:

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

- A : upprunalegur pakki, sem er fylki með lengd 31 og táknar bitana sem Kristinn sendi.
- Þessi stefja skilar spilltum pakka B , sem táknar bitana sem Þórhallur tekur á móti.
- Leyfilegt er að kalla í þessa stefju mest 100 sinnum í hverri fallbeitingu á `send_message`.

Seinni stefjan sem þú skalt útfæra er:

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

- R : fylki sem lýsir spilltu pökkunum. Pakkarnir eru upprunir frá pökkum sem Kristinn sendi í einu `send_message` kalli og eru gefnir í **sömu röð og þeir voru sendir** af Kristni. Sérhvert stak í R táknar fylki með lengd 31 sem táknar spilltan pakka.
- Þessi stefja skal skila fylki af S bitum sem er jafngilt upprunalega skilaboðinu M .
- Kallað er í þessa stefju oft í hverju prufutilviki, eða **nákvæmlega einu sinni** fyrir hvert samsvarandi `send_message` kall. **Röðin á** `receive_message` **stefjukallsetningum** er ekki endilega sú sama og röðin á samsvarandi `send_message` köllum.

Athugaðu að í yfirferðarkerfinu eru kallað á `send_message` og `receive_message` stefjurnar í **tveimur mismunandi forritum**.

Takmarkanir

- $1 \leq S \leq 1024$
- C er með nákvæmlega 31 stök, 16 þeirra eru 0 og 15 þeirra eru 1.

Stigagjöf

Ef köllin í `send_packet` fylgja ekki tilgreindu reglunum að ofan eða ef skilagildið úr einhverju kalli í `receive_message` er rangt fyrir eitthvert prufutilvik, þá fær lausnin þín 0 stig fyrir það prufutilvik.

Annars, látum Q vera hámarksfjölda kalla í stefjuna `send_packet` yfir allar fallbeitingar á `send_message` yfir öll prufutilvik. Látum X einnig vera jafnt:

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

Þá eru stigin reiknuð á eftirfarandi máta:

Hópur	Stig	Frekari takmarkanir
1	$10 \cdot X$	$S \leq 64$
2	$90 \cdot X$	Engar frekari takmarkanir.

Athugaðu að í sumum tilfellum getur yfirferðarforritið aðlagð sig að útfærslu þinni. Það þýðir að gildin sem `send_packet` skilar geta verið breytileg, ekki bara vegna inntaki kallsins, heldur vegna margra annarra breytna, eins og inntök og skilagildi fyrri kalla í stefjuna og talna úr gervislembitalnagjafa í yfirferðarforritinu. Yfirferðarforritið er **löggengt** að því leyti að ef þú keyrir það tvisvar og að í báðum keyrslum sendirðu nákvæmlega sömu pakkanna, mun það gera sömu breytingar á þeim.

Sýnidæmi

Íhugaðu eftirfarandi kall.

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

Skilaboðið sem Kristinn reynir að senda til Þórhalls er `[0, 1, 1, 0]`. Bitarnir með vísa 0 til 15 eru ekki undir stjórn Evu á meðan bitarnir með vísa 16 til 30 eru undir stjórn Evu.

Fyrir þetta sýnidæmi skulum við gera ráð fyrir að hegðun Evu sé að hún fyllir bitana í samliggjandi runu til skiptis með 0 og 1. Í öðrum orðum, hún stillir gildi fyrsta bitans sem hún stjórnar sem 0, á vísi 16, gildi annars bitans sem hún stjórnar sem 1, á vísi 17, gildi þriðja bitans sem hún stjórnar sem 0, á vísi 18, og svo framvegis.

Kristinn ákveður að senda tvo bita frá upprunalega skilaboðinu sínu í pakka á eftirfarandi máta: hann sendir fyrsta bitann í fyrstu 8 vísunum sem hann stjórnar og annan bitann í næstu 8 vísunum sem hann stjórnar.

Kristinn sendir því eftirfarandi pakka:

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

Athugaðu að Eva getur breytt bitunum á síðustu 15 vísunum, þannig Kristinn getur still þá eins og hann vill, því Eva gæti yfirskrifað þá. Með ályktun okkar um hegðun Evu skilar stefjan: $[0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0]$.

Kristinn ákverðu svo að senda síðustu tvo bitana á M í seinni pakkanum á svipaðan máta eins og áður:

```
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])
```

Með ályktun okkar um hegðun Evu skilar stefjan: $[1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0]$.

Kristinn getur sent fleiri pakka, en kýs að gera það ekki.

Yfirferðarforritið framkvæmir svo eftirfarandi kall:

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

Þórhallur endurheimtar skilaboðið M á eftirfarandi máta. Frá sérhverjum pakka tekur hann fyrsta bitann sem kemur fyrir tvisvar í röð og síðasta bitann sem kemur tvisvar í röð. Það þýðir að úr fyrsta pakkanum tekur hann bita $[0, 1]$ og úr seinni pakkanum tekur hann bita $[1, 0]$. Með því að setja þá saman, endurheimtar hann skilaboðið $[0, 1, 1, 0]$, sem er rétt skilagildi fyrir þetta kall í `receive_message`.

Það má sýna fram á að með ályktun okkar um hegðun Evu og fyrir skilaboð með lengd 4, þá virkar þessi aðferð fyrir Þórhall á réttan máta til að endurheimta M , sama hvert gildið á C er. Aðferðin er samt ekki rétt fyrir almenna tilvikið.

Sýnisyfirferðarforrit

Sýnisyfirferðarforritið aðlagar sig ekki að útfærslu þinni. Í staðin er hegðun Evu sú að hún fyllir samliggjandi runu af bitum með 0 og 1 til skiptis, alveg eins og er lýst í sýnidæminu að ofan.

Snið inntaks: **Fyrsta lína inntaks inniheldur heiltölu T , sem táknar fjölda atburðarása. T atburðarásir fylgja.** Sérhver þeirra er á eftirfarandi sniði:

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

Snið úttaks: Sýnisyfirferðarforritið skrifar niðurstöðuna á sérhverri atburðarás í sömu röð og þær koma fyrir í inntakinu á eftirfarandi sniði:

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

Hér er K fjöldi kalla í `send_packet`, D er skilaboðið sem `receive_message` skilar og L er lengdin.