

Message

Այժան և Բասման երկու ընկերներ են, որ իրար հաղորդագրություններ են ուղարկում։ Այժայի M հաղորդագրությունն իրենից ներկայացնում է S բիթերի (այսինքն զրոների և մեկերի) հաջորդականություն, որը նա ցանկանում է ուղարկել Բասմային։ Այժան Բասմայի հետ հաղորդակցվում է նրան ուղարկելով **փաթեթներ**։ Փաթեթը 31 հատ բիթերի հաջորդականությունը է, բիթերը համարակալված են 0-ից 30-ով։ Այժան ցանկանում է M հաղորդագրությունը հաղորդել Բասմային, դրա համար ուղարկելով նրան ինչ որ քանակությամբ փաթեթներ։

Դժբախտաբար, Կլեոպատրան խանգարեց Այժայի և Բասմայի միջև հաղորդակցությանը, նա ունակ է **փչացնել** փաթեթները։ Կա ճիշտ 15 ինդեքս, որոնցում գրված բիթը Կլեոպատրան կարող է փոփոխել յուրաքանչյուր փաթեթում։ Մասնավորապես, կա 31 երկարության C զանգված, որում յուրաքանչյուր տարր 0 կամ 1 է, և հետևյալ իմաստն ունի.

- $C[i] = 1$ նշանակում է, որ i ինդեքսով բիթը կարող է փոփոխվել Կլեոպատրայի կողմից։ Այդպիսի ինդեքսները կանվանենք՝ **հսկվող** Կլեոպատրայի կողմից։
- $C[i] = 0$ նշանակում է, որ Կլեոպատրան չի կարող փոխել i ինդեքսով բիթը։

C զանգվածը պարունակում է ճիշտ 15 հատ մեկ և 16 հատ զրո։ M հաղորդագրությունն ուղարկելու ողջ ընթացքում, Կլեոպատրայի կողմից հսկվող ինդեքսները նույնն են մնում բոլոր փաթեթների համար։ Այժան գիտի, թե որ 15 ինդեքսներն են հսկվում Կլեոպատրայի կողմից։ Բասման միայն գիտի, որ 15 հատ բիթ է հսկվում Կլեոպատրայի կողմից, բայց նա չգիտի, թե հատկապես որ ինդեքսներն են դրանք։

Դիցուք, Այժան որոշում է ուղարկել A փաթեթը (որը կանվանենք **օրիգինալ փաթեթ**)։ Դիցուք Բասման ստանում է B փաթեթը (որը կանվանենք **փչացած փաթեթ**)։ Յուրաքանչյուր i համար, որտեղ $0 \leq i < 31$

- եթե Կլեոպատրան չի կարողանում հսկել i ինդեքսով բիթը ($C[i] = 0$), Բասման ստանում է i բիթն այնպես, ինչպես Այժան այն ուղարկել է, այսինքն ($B[i] = A[i]$),
- հակառակ դեպքում, Կլեոպատրան հսկում է i ինդեքսով բիթը ($C[i] = 1$), այս դեպքում $B[i]$ -ի արժեքը Կլեոպատրան է որոշում։

Յուրաքանչյուր փաթեթն ուղարկելուց հետո, Այժան անմիջապես իմանում է, թե ինչպիսին է փչացած փաթեթը։

Այն բանից հետո, երբ Այշան ուղարկում է բոլոր փաթեթները, Բասման ստանում է փչացած փաթեթները **այն հերթականությամբ, ինչ հերթականությամբ որ դրանք ուղարկվել են**, և նա պետք է վերականգնի M հաղորդագրությունը:

Ձեր խնդիրն է մշակել և իրականացնել այնպիսի ռազմավարություն, որը թույլ կտա Այշային ուղարկել M հաղորդագրությունը Բասմային, այնպես որ Բասման կարողանա վերականգնել M -ը փչացած փաթեթների միջոցով: Մասնավորապես, դուք պետք է իրականացնեք երկու ֆունկցիա: Առաջին ֆունկցիան կատարում է Այշայի գործողությունը: Նրան տրվում է M հաղորդագրությունը և C զանգվածը, և այն պետք է ինչ-որ փաթեթներ ուղարկի Բասմային: Երկրորդ ֆունկցիան կատարում է Բասմայի գործողությունները: Նրան տրվում են փչացած փաթեթները, և այն պետք է վերականգնի M հաղորդագրությունը:

Իրականացման մանրամասներ

Առաջին ֆունկցիան, որ պետք է իրականացնեք, այսպիսի է.

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

- M . զանգված է, երկարությունը S է, նկարագրում է այն հաղորդագրությունը, որ Այշան պետք է ուղարկի Բասմային:
- C . ամբողջ թվերի զանգված, երկարությունը 31, ցույց է տալիս, թե որ բիթերն են հսկվում Կլեոպատրայի կողմից:
- Այս ֆունկցիան կարող է կանչվել **առավելագույնը 2100 անգամ** յուրաքանչյուր թեստի համար:

Այս ֆունկցիան կարող է անել հետևյալ ֆունկցիայի կանչեր, փաթեթն ուղարկելու համար.

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

- A . օրիգինալ փաթեթը (31 երկարության զանգված), որը ներկայացնում է Այշայի կողմից ուղարկված բիթերը:
- Այս ֆունկցիան վերադարձնում է փչացած B փաթեթը, որը ներկայացնում է Բասմայի ստացած բիթերը:
- Այս ֆունկցիան `send_message`-ի յուրաքանչյուր կանչի ժամանակ կարող է կանչվել **առավելագույնը 100 անգամ**:

Երկրորդ ֆունկցիան, որը դուք պետք է իրականացնեք, հետևյալն է.

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

- R . զանգված, որը նկարագրում է փչացած փաթեթները: Այս փաթեթները ստեղծվել են Այշայի կողմից ուղարկված փաթեթների հիման վրա `send_message`-ի մեկ կանչի ժամանակ և տրված են **այն կարգով, ինչ կարգով, որ դրանք ուղակել է** Այշան:
- Այս ֆունկցիան պետք է վերադարձնի S բիթերի զանգված, որը հավասար է M օրիգինալ հաղորդագրությանը:
- Այս ֆունկցիան կարող է կանչվել **բազմաթիվ անգամներ** յուրաքանչյուր թեստի համար, **ճիշտ մեկ անգամ** համապատասխան `send_message` կանչում: `receive_message` **ֆունկցիայի կանչերի հերթականությունը** պարտադիր չէ, որ նույնը լինի, ինչ `send_message` կանչերի հերթականությունն է:

Նկատենք, որ գնահատման համակարգում `send_message` և `receive_message` ֆունկցիաները կանչվում են **առանձին ծրագրերում**:

Սահմանափակումներ

- $1 \leq S \leq 1024$
- C -ն ունի ճիշտ 31 տարր, որոնցից 16-ը 0-ներ են, իսկ 15-ը 1-եր են:

Ենթախնդիրներ և Գնահատում

Եթե թեստերից որևէ մեկում, `send_packet` ֆունկցիայի կանչերը չեն բավարարում վերը նշված պահանջներին, կամ `receive_message` ֆունկցիայի կանչերից թեկուզ մեկում վերադարձի արժեքը կոռեկտ չէ, ձեր լուծումը ստանում է 0 միավոր:

Հակառակ դեպքում, դիցուք, եթե դիտարկենք բոլոր թեստերում Q `send_message` կանչերում `send_packet` ֆունկցիայի կանչերի մաքսիմալ քանակը Q է, Նաև դիցուք X -ը հավասար է

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

Այդ դեպքում միավորը հաշվվում է հետևյալ կերպ.

Ենթախնդիր	Միավոր	Լրացուցիչ սահմանափակումներ
1	$10 \cdot X$	$S \leq 64$
2	$90 \cdot X$	Լրացուցիչ սահմանափակումներ

Նկատենք, որ որոշ թեստերում գրեյդերի պահվածքը **հարվարվող** է: Դա նշանակում է, որ `send_packet` -ի վերադարձրած արժեքները կարող են կախված լինել նրա մուտքային արգումենտներից և այդ ֆունկցիայի նախքին կանչերի վերադարձրած արժեքներից:

Օրինակ

Դիտարկենք հետևյալ կանչը.

```
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])
```

Այշան փորձում է Բասմային ուղարկել հետևյալ հաղորդագրությունը. [0,1,1,0]: 0-ից 15 բիթերը Կլեոպատրան չի կարող փոխել, փոխարենը կարող է փոխել 16-ից 30 բիթերը:

Միայն այս օրինակում համարաենք, որ Կլեոպատրայի պահվածքը դետերմինացված է, այսինքն միարժեք է, և նա իր վերահսկողության տակ գտնվող բիթերը լցնում է մեկումեջ 0-ներով և 1-երով, այսինքն նա իր վերահսկողության տակ գտնվող առաջին ինդեքսով տարրին վերագրում է 0 (մեր դեպքում դա 16 ինդեքսն է), իր վերահսկողության տակ գտնվող երկրորդ տարրին (դա 17 ինդեքսն է) վերագրում է 1, իր վերահսկողության տակ գտնվող երրորդ տարրին (դա 18 ինդեքսն է) վերագրում է 0, և այդպես շարունակ:

Այշան կարող է որոշել սկզբնական փաթեթի երկու բիթերն ուղարկել հետևյալ կերպ. նա առաջին բիթը կուղարկի իր հսկողության տակ գտնվող առաջին 8 ինդեքսներում, իսկ երկրորդ բիթը, հաջորդ 8 ինդեքսներում:

Այսպես Այշան ուղարկելու համար ընտրում է հետևյալ փաթեթը.

```
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])
```

Նկատենք, որ Կլեոպատրան կարող է փոխել միայն վերջին 15 ինդեքսները, այսպիսով Այշան կարող է դրանց կամայական արժեք տալ, քանի որ դրանք կարող են փոխվել: Այս ռազմավարության դեպքում Կլեոպատրայի ֆունկցիան վերադարձնում է. [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].

Այշան որոշում է M -ի վերջին երկու բիթերն ուղարկել երկրորդ փաթեթով, վերը նկարագրված եղանակի պես.

```
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])
```

Կլեոպատրայի ենթադրյալ ռազմավարության դեպքում, ֆունկցիան վերադարձնում է. [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].

Այշան կարող է էլի փաթեթներ ուղարկել, բայց նա նախընտրում է չանել:

Հետո գրեյդերը կատարում է հետևյալ կանչը.

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

Բասման M հաղորդագրությունը վերականգնում է հետևյալ կերպ. Յուրաքանչյուր փաթեթից նա ընտրում է առաջին բիթը, որը իրար հետևից երկու անգամ է հանդիպում, և վերջին բիթը, որը կրկին իրար հետևից երկու անգամ է հանդիպում: Հետևաբար, առաջին փաթեթից նա վերցնում է $[0, 1]$ բիթերը, և եկրորդ փաթեթից նա վերցնում է $[1, 0]$ բիթերը: Դրանք իրար կողք դնելով նա վերականգնում է հաղորդագրությունը. $[0, 1, 1, 0]$, որը կոռեկտ վերադարձի արժեք է `receive_message` կանչի համար:

Կարելի ցույց տալ, որ Կլեոպատրայի ենթադրյալ ռազմավարության և 4 երկարության հաղորդագրությունների դեպքում, Բասմայի այս մոտեցումը ճիշտ վերականգնում է M -ը, անկախ C -ի արժեքից: Սակայն, ընդհանուր դեպքում սա ճիշտ չէ:

Գրեյդարի նմուշ

Գրեյդերի նմուշը հարմարվող չէ: Փոխարենը, Կլեոպատրայի պահվածքը դետերմինացված է, այսինքն միարժեք է, և նա լցնում է իր վերահսկողության տակ գտնվող բիթերը մեկումեջ 0-ներով և 1-երով, ինչպես նկարագրված է վերևի օրինակում:

Մուտքային տվյալներ. **Առաջին տողը պարունակում է մի T ամբողջ թիվ, որը նկարագրում է սցենարների քանակը:** Ապա հաջորդում են T սցենարներ: Նրանցից յուրաքանչյուրը տրվում է հետևյալ ձևաչափով.

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

Ելքային տվյալներ. Գրեյդերի նմուշը գրում է T սցենարներից յուրաքանչյուր արդյունքը նույն հերթականությամբ ինչես նրանք տրված են մուտքային տվյալներում, հետևյալ ձևաչափով.

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

Այստեղ, K -ն `send_packet` ֆունկցիայի կանչերի քանակն է, D -ն `receive_message`-ի վերադարձրած հաղորդագրությունն է, նրա երկարությունը L է: