

1. Ուղղի ներկում

Ուղղի ինտերվալ ասելով կհասկանանք $[a, b)$ հատվածը, որին ձախ a ծայրակետը պատկանում է, իսկ աջ b ծայրակետը չի պատկանում: $[0, 1000000000)$ ինտերվալը ներկել են սպիտակ գույնով: Հետո կատարել են N ներկելու գործողություն: Յուրաքանչյուր գործողության ժամանակ տրվում է ինչ-որ ինտերվալ, այդ ինտերվալի ներսում գտնվող ինտերվալները փոխում են իրենց գույնը հակառակ գույնի (սևը սպիտակի, սպիտակը սևի): Հարկավոր է գրել ծրագիր, այդ բոլոր գործողություններից հետո ամենամեծ սպիտակ ինտերվալի երկարությունը գտնելու համար:

Մուտքը

Առաջին տողում տրված է N ($1 \leq N \leq 100$) ինտերվալների քանակը: Հաջորդ N տողերից յուրաքանչյուրում տրված է հերթական ինտերվալի ծայրակետերի կոորդինատները:

Ելքը

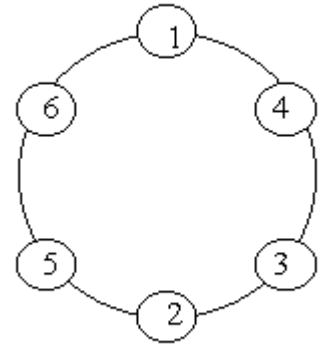
Հարկավոր է արտածել մեկ թիվ՝ ամենաերկար սպիտակ ինտերվալի երկարությունը:

Մուտքային և ելքային տվյալների օրինակ

| Մուտք | Ելք |
|--|-----|
| 4 20 50 10 35 40 90 100 1000000000 | 15 |

2. Պարզ թվերի օղակ

Տրված n բնական զույգ թվի համար հարկավոր է կազմել պարզ թվերի օղակ: Օղակը պետք է կազմել n հատ շրջաններից ինչպես ցույց է տրված նկարում: Այդ շրջանների ներսում 1-ից n թվերը պետք է տեղադրել այնպես, որ երկու հարևան թվերի գումարը լինի պարզ թիվ: Ուշադրություն, առաջին շրջանում միշտ պետք է լինի 1 թիվը: Գրեք ծրագիր, որը կառուցի այդպիսի օղակներ:



Մուտք

Մուտքում տրված է մի n ($0 < n \leq 12$) բնական թիվ:

Ելք

Ելքային տվյալների տեսքը պետք է լինի այնպես, ինչպես ցույց է տրված օրինակում: Յուրաքանչյուր տողում պետք է տրված լինի շրջաններում գրված թվերի մի հաջորդականություն ժամ պաթի կամ հակառակ ուղղությամբ: Հաջորդականության ամեն թվից (նաև վերջին թվից) հետո պետք է լինի մեկ բացակ: Համարակալումը միշտ պետք է սկսվի 1-ից: Հարկավոր է արտածել բոլոր հնարավոր լուծումները: Կրկնություն չպետք է լինի: Լուծումները պետք է դասավորված լինեն բառարանային կարգով: Դա նշանակում է, որ i -րդ և $i+1$ -րդ հաջորդականությունների համար պետք է գոյություն ունենա այնպիսի մի k թիվ, որ երկու հաջորդականությունների առաջին k անդամներն իրար հավասար են, սակայն i -րդ հաջորդականության $k+1$ -րդ անդամը մեծ է $i+1$ -րդ հաջորդականության $k+1$ -րդ անդամից:

Մուտքային և ելքային տվյալների օրինակներ

| Մուտք | Ելք |
|-------|--|
| 6 | 1 4 3 2 5 6 1 6 5 2 3 4 |
| 8 | 1 2 3 8 5 6 7 4 1 2 5 8 3 4 7 6 1 4 7 6 5 8 3 2 1 6 7 4 3 8 5 2 |

3. Չնջող ծրագիր

Կարենը համակարգչային էքսպերտ է: Նա առաջադրանք է ստացել ստեղծել տվյալները ջնջող ծրագրային ապահովում: Շատ կարևոր է, որ տվյալներն այնպես ջնջվեն, որ հետո հնարավոր չլինի դրանք վերականգնել: Համացանցում համապատասխան ծրագիր չգտնելով Կարենը ինքը գրեց ծրագիրը: Ծրագրի ինտերֆեյսը պարզ է. այն հարցնում է ջնջվող ֆայլի անունը և մի n թիվ, որը ցույց է տալիս, թե տվյալները ջնջելու գործողությունը քանի անգամ է պետք կատարել: Այդ թիվը պատկանում է 1-ից (արագ ջնջում) 20 (մաքսիմում ապահովություն) տիրույթին: Կարենի ծրագիրը անցնում է ֆայլի վրայով բայթ առ բայթ և յուրաքանչյուր 0 բիթը դարձնում է 1, իսկ 1-ը դարձնում է 0:

Կարենը դիմել է ձեզ որպես անկախ թեստավորողի: Դուք պետք է գրեք ծրագիր, որը ստուգում է Կարենի գրած ծրագրի աշխատանքը:

Մուտք

Առաջին տողում տրված է մի $1 \leq N \leq 20$ ամբողջ թիվ: Հաջորդ երկու տողերը բաղկացած են 0-ներից և 1-երից: Դրանցից առաջինը ներկայացնում է ֆայլի բիթերը ջնջելու պրոցեսը սկսելուց առաջ, իսկ երկրորդը՝ այդ պրոցեսն ավարտելուց հետո: Այդ երկու տողերի երկարություններն իրար հավասար են և չեն գերազանցում 1000-ը:

Ելք

Ելքում պետք է արտածել “Deletion succeeded” արտահայտությունը, եթե բիթերից յուրաքանչյուրը N անգամ փոխվել է, ինչպես նախատեսված է Կարենի ծրագրում: Հակառակ դեպքում պետք է արտածել “Deletion failed” արտահայտությունը:

Օրինակներ

| Մուտք | Ելք |
|---|--------------------|
| 1 10001110101000001111010100001110 01110001010111110000101011110001 | Deletion succeeded |
| 20 0001100011001010 0001000011000100 | Deletion failed |

4. Մուտացիաներ

Տրված է 5n հաստ թվերի հաջորդականություն A_1, A_2, \dots, A_{5n} . Տրված են նաև երեք տիպի հարցումներ.

1. “ ? x y “

Այստեղ x-ն ու y-ն այնպիսի ամբողջ թվեր են, որոնց համար $1 \leq x \leq y \leq 5n$: Գտեք x-ից մինչև y դիրքերը ներառյալ թվերի գումարը $A_x + A_{x+1} + \dots + A_y$:

2. “ < x “

Այստեղ x-ն այնպիսի ամբողջ թիվ է, որ $1 \leq x \leq 5n-1$: Հաջորդականությունը x անգամ ցիկլիկ տեղաշարժեք դեպի ձախ, այսինքն, A հաջորդականությունը փոխարինեք $A_{x+1}, A_{x+2}, \dots, A_{5n}, A_1, \dots, A_x$ -ով:

3. “ # P “

Այստեղ P-ն 5 երկարությամբ տեղափոխություն է: Տրոհեք այդ հաջորդականությունը 5-տարրանոց սեգմենտների P տեղափոխությունը կիրառեք այդ սեգմենտներից յուրաքանչյուրի նկատմամբ: Յուրաքանչյուր $A_{5k+1}, A_{5k+2}, \dots, A_{5k+5}$ սեգմենտը փոխարինված է $A_{5k+p1}, A_{5k+p2}, \dots, A_{5k+p5}$ -ով:

Մուտք

Մուտքի առաջին տողը պարունակում է մեկ ամբողջ թիվ՝ n ($1 \leq n \leq 2 \times 10^4$): Երկրորդ տողը պարունակում է 5n ամբողջ թվեր A_i ($|A_i| \leq 10^4$): Երրորդ տողը պարունակում է հարցումների m ($0 \leq m \leq 10^5$) քանակը: Հաջորդ m տողերից յուրաքանչյուրը պարունակում է վերը նկարագրված ձևաչափով մեկ հարցում:

Ելք

Յուրաքանչյուր ? հարցման համար արտածեք գումարը առանձին տողով:

Օրինակ

| Մուտք | Ելք |
|----------------------|-----|
| 2 | 55 |
| 1 2 3 4 5 6 7 8 9 10 | 35 |
| 10 | 13 |
| ? 1 10 | 7 |
| < 3 | 42 |
| ? 2 6 | 13 |
| # 3 1 2 5 4 | |
| ? 3 4 | |
| ? 5 5 | |
| < 8 | |
| ? 2 9 | |
| # 5 4 3 2 1 | |
| ? 5 6 | |

5. Անսարք հեռուստացույց

Դավիթն ունի մի շատ հին հեռուստացույց, որի որոշ կոճակներ չեն աշխատում: Որևէ կոճակի սեղմումը (եթե հեռուստացույցը նոր է) դուրս է մղում մյուս բոլոր կոճակները՝ սեղմած թողնելով միայն մեկ կոճակ: Այժմ կոճակի սեղմումը դուրս է մղում որոշ կոճակներ, մյուս կոճակների դիրքերը թողնելով անփոփոխ: Դավիթը գիտի ամեն կոճակը սեղմելուց մյուս որ կոճակների դիրքերն են փոխվում:

Գրել ծրագիր, որը կօգնի Դավիթին գտնել սեղմումների ամենակարճ հաջորդականությունը, որից հետո սեղմված է մնում միայն երրորդ կոճակը:

Մուտքը

Մուտքի առաջին տողը պարունակում է կոճակների N ($3 \leq N \leq 20$) քանակը: Երկրորդ տողը պարունակում է, իրարից մեկական բացակով անջատված, N 0-ներ և 1-եր, որոնք ցույց են տալիս կոճակների սկզբնական դիրքերը: 0-ն նշանակում է, որ կոճակը սեղմված չէ, իսկ 1-ը՝ որ այն սեղմված է: Հաջորդ N տողերում տրված է, թե որ կոճակներն են դուրս մղվում համապատասխան կոճակը սեղմելիս: $(M+2)$ -րդ տողը սկսվում է K թվով, որին հետևում են K թվեր (դասավորված աճման կարգով): Կոճակները համարակալված են 1-ից N թվերով: Կոճակը չի կարող ինքն իրեն դուրս մղել: Կոճակը կարող է ոչ մի կոճակ դուրս չմղել:

Մուտքային տվյալներն այնպիսին են, որ լուծում գոյություն ունի:

Ելքը

Ելքի առաջին և միակ տողում պետք է արտածել կոճակների համարների ամենակարճ հաջորդականության երկարությունը ըստ որի կոճակների սեղմելուց միայն երրորդ կոճակն է մնում սեղմված:

Մուտքային և ելքային տվյալների օրինակներ

| Մուտք | Ելք |
|---|-----|
| 3 1 1 0 2 2 3 2 1 3 2 1 2 | 1 |
| 4 0 1 0 1 3 2 3 4 1 1 1 1 0 | 2 |
| 5 1 1 0 0 1 4 2 3 4 5 4 1 3 4 5 2 2 4 0 4 1 2 3 4 | 3 |

6. Կոնյակի արտադրություն

Կոնյակի գործարանում ցանկանում են պլանավորել կոնյակի արտադրություն: Ամեն օր ճիշտ մեկ տակառ պետք է լցվի կոնյակով: Յուրաքանչյուր տակառ որոշակի քանակությամբ օրեր հետո պետք է բացվի: Այդ թիվը փակցվում է տակառի վրա: Յուրաքանչյուր տակառ միայն մեկ անգամ է օգտագործվում: Հարկավոր է գրել ծրագիր, որը հաշվի, թե տակառներում կոնյակ լցնելու պլանավորումը ճիշտ կազմակերպելու դեպքում վերջին տակառը ամենաշուտը քանի օր հետո կբացվի:

Մուտք

Առաջին տողում տրված է տակառների N ($1 \leq N \leq 100000$) քանակը: Հաջորդ տողում տրված են N հատ թվեր, որոնցից i -րդը ցույց է տալիս, թե i -րդ տակառում կոնյակը քանի օր պետք է մնա մինչև բացելը: Այդ թվերը դրական ամբողջ թվեր են և չեն գերազանցում 1000000-ը:

Ելք

Պետք է արտածել մեկ թիվ, որը պետք է ցույց տա, թե առաջին տակառը փակելուց հետո մինիմումը քանի օր անց պետք է բացել վերջին տակառը:

Մուտքային և ելքային տվյալների օրինակներ

| Մուտք | Ելք |
|-----------------------|-----|
| 4 2 3 4 3 | 6 |
| 6 39 9 38 39 22 35 | 41 |

7. Բաց տիրույթներ

Հարթության մեջ տարված ուղիղները հարթությունը տրոհում են որոշակի քանակությամբ փակ և բաց տիրույթների: Հարկավոր է գրել ծրագիր, որը հաշվի, թե առնվազն քանի ուղիղ է պետք ավելացնել, որպեսզի բաց տիրույթների քանակը հավասար լինի ոչ պակաս քան տրված N թիվը:

Մուտք

Առաջին տողում տրված են ուղիղների M քանակը և N թիվը ($1 \leq M, N \leq 100$): Հաջորդ N տողերից յուրաքանչյուրը պարունակում է չորս ամբողջ թիվ՝ x_1, y_1, x_2, y_2 , որոնք բացարձակ արժեքով չեն գերազանցում 10000-ը: Սա նշանակում է, որ հարթության մեջ տարված է (x_1, y_1) և (x_2, y_2) կետերով անցնող ուղիղ: Այդ երկու կետերը միշտ իրարից տարբեր են:

Ելք

Պետք է արտածել մեկ թիվ, որը ցույց տա, թե առնվազն քանի ուղիղ պետք է ավելացնել, որպեսզի բաց տիրույթների քանակը լինի N -ից մեծ կամ հավասար:

Մուտքային և ելքային տվյալների օրինակներ

| Մուտք | Ելք |
|--------------------------------------|-----|
| 2 1 1 1 -2 0 | 0 |
| 5 3 0 5 5 5 0 0 1 1 2 2 3 3 | 1 |

8. Լեռնադահուկորդներ

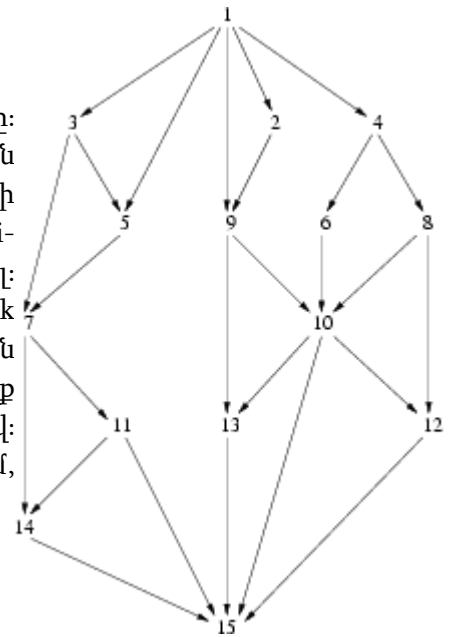
Լեռնադահուկորդների թիմը մարզումներ է անցկացնում Բայթասարում: Կա մի ճոպանուղի, որը տանում է մինչև սարի հյուսիսային գագաթը: Բոլոր սահքուղիները ձգվում են ճոպանուղու վերևի կանգառից մինչև ներքևի կանգառը: Մարզիկներն իրար պիտի հանդիպեն միայն այս երկու կետերում: Բոլոր սահքուղիները վերևից ներքև են տանում:

Սահքուղիների քարտեզը բաղկացած է բացատներից, որոնք իրար միացված են սահքուղիների ցանցով: Բացատները գտնվում են տարբեր բարձրության վրա: Երկու բացատ ամենաշատը մի սահքուղով կարող են միացված լինել: Սահքուղիները հատվում են միայն բացատներում:

Գրել ծրագիր, որը ներածում է սահքուղիների քարտեզը և պարզում է, թե ամենաշատը քանի մարզիկ կարող է մասնակցել մարզումներին:

Մուտք

Առաջին տողում տրված է բացատների n ($2 \leq n \leq 5\,000$) քանակը: Հաջորդ $n-1$ տողերից յուրաքանչյուրում տրված է մեկական բացակով իրարից բաժանված ամբողջ թվերի հաջորդականություն: $(i+1)$ -րդ տողի թվերը նկարագրում են, թե i -րդ բացատից ցած սահելով որ բացատներին կարելի է հասնել: Տողի առաջին k թիվը այդ բացատների քանակն է, իսկ հաջորդ k թվերը այդ բացատների համարներն են, որոնք կարգավորված են ըստ դեպի իրենց տանող սահքուղիների արևմուտքից արևելք ուղղությամբ: Բացատները համարակալված են 1-ից n թվերով: Ճոպանուղու վերևի կանգառը գտնվում է համար 1 բացատում, իսկ ներքևինը՝ համար n -ում:



Ելք

Պետք է արտածել ճիշտ մեկ թիվ, որը պետք է ցույց տա, թե առավելագույնը քանի մարզիկ է կարող մասնակցել մարզումներին:

Մուտքային և ելքային տվյալների օրինակ

| Մուտք | Ելք |
|-------------|-----|
| 15 | 3 |
| 5 3 5 9 2 4 | |
| 1 9 | |
| 2 7 5 | |
| 2 6 8 | |
| 1 7 | |
| 1 10 | |
| 2 14 11 | |
| 2 10 12 | |
| 2 13 10 | |
| 3 13 15 12 | |
| 2 14 15 | |
| 1 15 | |
| 1 15 | |
| 1 15 | |