Task 1. Monopoly

Դենին շատ է սիրում խաղալ մոնոպոլիա խաղը: Բայց մեկ խաղի տևողությունը շատ երկար է՝ 5-ից 7 ժամ: Դրա համար Դենին սկսում է մտածել դասական կանոնները փոխելու մասին: Մոնոպոլիայում խաղացողները սովորաբար մի ուղղությամբ շարժվում են տեղից տեղ և որոշ ժամանակ անց վերադառնում են սկզբին, նորից սկսում և այլն: Նոր տարբերակում շարժումը կրկին կլինի տեղից տեղ, սակայն ընթացիկ տեղից կարող է լինել մի քանի հնարավորություն հաջորդ քայլի համար: Դենին ցանկանում է գտնել այնպիսի ուղղորդված կապեր տեղերի միջև, որպեսզի խաղացողը երբեք չվերադառնա այն տեղը, որտեղ եղել է, անկախ նրանից, թե ինչպես է նա շարժվում (իհարկե, հետևելով կանոններին): Այսպիսով, խաղն ավելի կարճ կտևի:

Նա արդեն սկսել է կառուցել նոր դաշտը՝ նա ընտրել է N քանակով տեղեր (տեղերը համարակալված են 1-ից ***N*** թվերով), և նա պատրաստել է ***M*** կապերի ցուցակ (բոլոր կապերն ունեն ուղղություն, և չկա կապ, որը տեղը կապի ինքն իր հետ): Եթե *i* տեղը կապված է *j* տեղի հետ, ապա հակառակ ուղղությամբ ուղիղ, այսինքն j-ից I, կապ գոյություն չունի, նաև, i-ից j այլ ուղիղ կապեր գոյություն չունեն: Դենին մտածեց, որ նոր դաշտը պատրաստ է, բայց հանկարծ նկատեց, որ իր ուզած պայմանը (երբ տեղից տեղ ես տեղափոխվում, օգտագործելով կապերը, չես կարող վերադառնալ նախկինում այցելած տեղ) չի համապատասխանում իր ցանկին: Նա նախ մտածեց հեռացնել որոշ ուղիղ կապեր, բայց այդ դեպքում ստիպված կլինի ցուցակը նորից գրել, որը կարող է շատ երկար տևել: Ահա թե ինչու Դենին որոշեց շրջել որոշ ուղղորդված կապերի ուղղությունը:

Խնդիրը

Դուք պարբերաբար Դենիի հետ մոնոպոլիա եք խաղում: Ահա թե ինչու ցանկանում եք օգնել նրան գրելով **monopoly** ծրագիրը, որը կհուշի նրան, թե որ կապերի ուղղությունը շրջի, որ նշված պայմանը տեղի ունենա: Ծրագիրը պետք է պարունակի *find_reverse* ֆունկցիան, որը կկոմպիլացվի ժյուրիի ծրագրի հետ:

Իրականացման մանրամասներ

find_reverse ֆունկցիան պետք է ունենա հետևյալ նախատիպը.

std::string find_reverse (int N, int M, int connections[][2]);

Ժյուրիի ծրագրի կողմից այն կանչվում է միայն մեկ անգամ՝ երեք պարամետրով՝ **N** – նոր դաշտու�տեղերի քանակը, **M** – ուղղորդված կապերի քանակը Դենիի ցուցակում և connections զանգվածը, որն ունի M տող, որոնցից յուրաքանչյուրը բաղկացած է երկու թվից՝ x և y, ուղղորդված կապի սկզբի և վերջի տեղերը: Այս ֆունկցիան պետք է վերադարձնի **M** երկարությամբ երկուական տող՝ ըստ connections զանգվածի կապերի հերթականության, յուրաքանչյուր կապի համար տողի մեջ պետք է դնել «1», եթե կապը պետք է շրջվի, և «0», հակառակ դեպքում: Եթե կա մեկից ավելի լուծում, կարող եք վերադարձնել դրանցից որևէ մեկը:

Դուք պետք է համակարգին ուղարկեք **monopoly.cpp** ֆայլը, որը պարունակում է *find_reverse* ֆունկցիայի իրականացումը: Ֆայլում կարող են ուրիշ ֆունկցիաներ և կոդեր նույնպես լինել, որոնք անհրաժեշտ են ձեր ծրագրին, բայց այն **չպետք է պարունակի** main ֆունկցիան: Նաև դուք չպետք է փորձեք կարդալ ստանդարտ մուտքից, ոչ էլ գրել ստանդարտ ելքում:

Սահմանափակումներ

- ♣ $3 \le \boldsymbol{N} \le 5 \times 10^5$
- ♣ $3 \le \boldsymbol{M} \le 1.5 \times 10^6$

Ենթախնդիրներ

Ենթախնդիր	Միավոր	***N***	***M***	Լրացուցիչ սահմանափակումներ
1	0	–	–	Օրինակը
2	15	≤ 7	≤ 21	–
3	40	$\le 10^3$	$\le 5 \times 10^3$	–
4	25	$\le 10^5$	$\le 5 \times 10^5$	–
5	20	$\le 5 \times 10^5$	$\le 1.5 \times 10^6$	–

Տվյալ ենթախնդրի համար նախատեսված միավորը ստանալու համար ձեր ծրագիրը պետք է անցկացնի այդ ենթախնդրի բոլոր թեստերը:

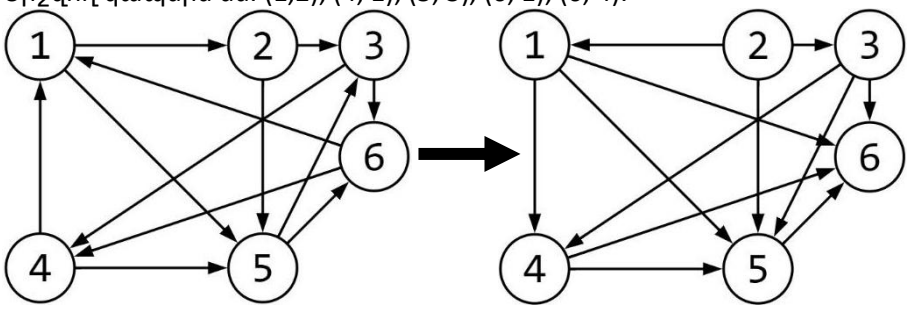
Լոկալ թեստավորում

Լոկալ թեստավորման համար ձեզ տրված է **Lgrader.cpp** ֆայլը: Դուք պետք է այն տեղադրեք ձեր **monopoly.cpp** ֆայլի հետ նույն ֆոլդերում, ավելացնեք նույն պրոյեկտին և պետք է իրար հետ կոմպիլացնեք **Lgrader.cpp** և **monopoly.cpp** ֆայլերը: Այդպիսով դուք կստեղծեք ծրագիր ձեր ֆունկցիայի աշխատանքը ստուգելու համար: Ծրագիրը պահանջելու է ստանդարտ մուտքից ներածել տվյալների հետևյալ հաջորդականությունը.

- Առաջին տողում երկու ամբողջ թվեր՝ մոնոպոլիայի դաշտում տեղերի ***N*** քանակը և կապերի ***M*** քանակը:
- Հաջորդ ***M*** տողերում. յուրաքանչյուր տողում պետք է լինեն երկու ամբողջ *x* և *y* թվեր՝ հերթական ուղղորդված կապի սկզբնական տեղը և վերջնական տեղը:

Ծրագիրը կարտածի գտնված երկուական տողը:

Լոկալ թեստավորման օրինակ

Մուտք	Ելք	Բացատրություն
6 12 1 2 1 5 2 3 2 5 3 4 3 6 4 1 4 5 5 3 5 6 6 1 6 4	10000010101 1	Շրջվող կապերն են. (1,2), (4, 1), (5, 3), (6, 1), (6, 4).  Վերևի նկարում պատկերված են մոնոպոլիայի տեղերը նախ Դենիի ցուցակում նշված ուղղորդված կապերով, ապա, ելքի ‘1’-երին համապատասխանող կապերի շրջումից հետո: Տեսնում ենք, որ սկզբում օգտագործելով (1, 2), (2, 3), (3, 4) և (4, 1) կապերը կարող ենք սկսել 1 տեղից և շարժվելով այդ ուղղորդված կապերով վերադառնալ նույն տեղը: Կարելի է տեսնել, որ արդյունքում ստացված դաշտում Դենիի պայմանը տեղի ունի: Նկատենք, որ ուրիշ թույլատրելի լուծումներ էլ կան. (4, 1), (5, 3), (6, 1), (6, 4) (1, 5), (2, 3), (2, 5), (3, 4), (3, 6), (4, 5), (5, 6)