```
Tito policiste porusuji smernici:
(4,8)(8,7)(12,6)
(5,5)(8,7)(14,11)(20,15)
(12,6)(14,11)(18,21)

Smernice byla dodrzena.

Tito policiste porusuji smernici:
(5,5)(10,11)(20,23)
(5,5)(15,11)(20,14)(25,17)
```

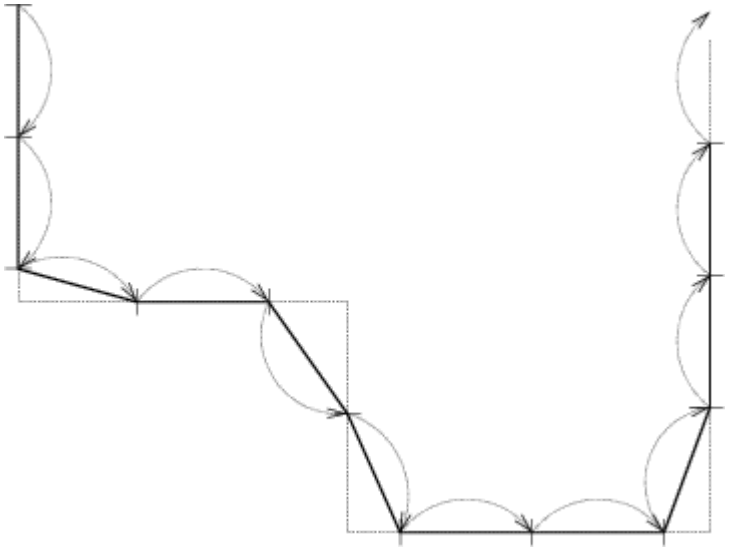


Počítáme ponorky

ponork.c, ponork.p, ponork.C
ponork.in, ponork.out,

Protoľe hlavní část zasedání se konala v Kongresovém Finančním Centru (KFC), bylo samozřejmě nutné zajistit tomuto místu kvalitní ochranu. Jak jste jistě sami viděli, přes den byla tato stavba hlídána více neľ dostatečně. Nás ale bude zajímat poněkud vąednějąí noc, kdy i větąina demonstrantů jiľ ulehla ke spánku.

Je jasné, ľe i v noci je třeba budovu ohlídat. Bývá k tomu určeno několik speciálně ąkolených stráľných, kteří obchází KFC celou noc po různých trasách. Nejdůleľitějąí z nich vede po úzké zídce kolem celé budovy. Tato trasa byla vytyčena ze strategických důvodů, nebo» z ní je velmi dobrý výhled do celého okolí. A aby bylo celé okolí pokryto, musí být na tuto trasu nasazeno stráľných několik. Ale protoľe lidí na práci je vľdy méně, neľ by bylo třeba, vedení Policie potřebuje určit minimální počet stráľných tak, aby stále měli celé okolí budovy v dohledu. K řeąení tohoto problému je vąak třeba znát, jak je trasa dlouhá a za jak dlouho ji stráľný projde. A to právě bude vaąím úkolem.



Pro určení délky trasy se u Policie pouľívá měření tzv. *policejních normokroků* (zkratka *ponork*). Zídka má podobu rovných úseků, které na sebe navazují v pravém úhlu. Kaľdý úsek má délku nejméně jeden ponork. Pro účely měření je v policejních předpisech stanoveno, ľe stráľný jdoucí po zídce nezkracuje při změně směru krok, ale překračuje rohy tak, ľe délka jeho kroku je stále přesně jeden ponork. ©ířku zídky povaľujeme za nulovou. Pokud se celá trasa nedá projít tak, ľe posledním krokem stráľný skončí přesně v koncovém bodě zídky, počítá se zbytek jako celý krok pouze v případě, ľe je dlouhý nejméně půl ponorku. Vaąím úkolem je určit délku trasy, která vede po zídce, v ponorcích. (Ilustrační obrázek odpovídá prvnímu z příkladů vstupu.)

Specificace vstupu:

První řádek vstupního souboru obsahuje celé kladné číslo Z, za kterým následuje postupně Z zadání. Kaľdé zadání začíná řádkem obsahujícím dvě hodnoty: K a U, ($1 <= K<1000$, $1 <=$

U<5000) kde K je délka kroku stráľného (velikost ponorku) a U je počet úseků zídky. Po nich následuje U čísel určujících délky jednotlivých úseků, kaľdé číslo je na samostatném řádku.

Specifikace výstupu:

Pro kaľdé zadání vypíąe program právě jeden řádek s textem "`Strazny ujde X ponorku.`" kde X je počet ponorků potřebných k projití celé zídky.

Příklad vstupu:

```
3
4 5
9
10
7
11
15
1 3
5
3
7
4 2
6
7
```

Příklad výstupu:

```
Strazny ujde 12 ponorku.
Strazny ujde 15 ponorku.
Strazny ujde 3 ponorku.
```



Rozejděte se!

rozchod.c, rozchod.p, rozchod.C
rozchod.in, rozchod.out,

Nejoblíbenějąí větou kaľdého správného policisty je „Občané, rozejděte se! Není tady nic k vidění!". Ovąem samotná věta vľdy nepomůľe. Existují obsáhlé příručky, které popisují, jak se má policista zachovat, pokud občané neuposlechnou dobře míněné výzvy a nerozejdou se. Nejdůleľitějąí rada pro takovou chvíli říká, ľe je dobré hlouček lidí rozdělit tak, aby jich bylo pohromadě co nejméně. A o takové dělení nám půjde i v této úloze. Úkolem bude hledat různá dělení hloučku, aby si z nich policista mohl vybrat to nejsnáze realizovatelné.

Pro zjednoduąení uvaľujeme pouze takzvané „ortogonální normohloučky stejnoměrně rozlehlých občanů", coľ je termín zavedený v dnes jiľ klasické příručce na toto téma *Rozcházení se snadno a rychle*. Takový normohlouček je moľné modelovat jako čtvercový útvar sloľený z N x N čtvercových občanů stejné půdorysové velikosti. Úkolem policisty je tento hlouček rozdělit na dvě poloviny. A ideální poloviny jsou samozřejmě takové, které jsou obě stejné, a to nejen co do velikosti, ale i do tvaru.

Kromě toho je vąak také nutné uvaľovat způsob oddělení obou polovin od sebe. Existuje speciální oddělovací policejní páska určená speciálně k tomuto účelu, v policejním ľargonu se jí říká „oddělovačka". Tuto páskou je moľné rychle ovinout kolem skupiny občanů, a tak je oddělit od zbytku davu. Vzhledem k tomu, ľe oddělovačka je velmi drahá, nesmí se nikdy stříhat ani krátit. Je proto nutné najít takové rozdělení normohloučku, kdy kaľdá ze vzniklých polovin má přesně daný obvod.

Vaąím úkolem je určit počet různých způsobů, kterými je moľné rozdělit čtvercový normohlouček o velikosti strany N takovým způsobem, aby vznikly dvě části o přesně daném obvodu M. Tyto části musí být stejné, to znamená, ľe jedna musí být převeditelná na druhou

pouze pomocí operací posunu, otáčení a zrcadlení. Dělící linie může samozřejmě vést pouze po hranách „jednotkových čtverečků", aby žádný občan neutrpěl zranění.

Specificace vstupu:

První řádek vstupního souboru obsahuje celé kladné číslo *Z*, za kterým následuje postupně *Z* zadání. Každé zadání sestává ze dvou čísel *N* a *M*, 2 <= *N* <= 20, 2 <= *M* <= 60, uvedených na stejném řádku a oddělených mezerou. První číslo (*N*) udává velikost strany čtverce, druhé (*M*) je délka pásky, tedy obvod každé z polovin, které mají rozdělením vzniknout.

Specifikace výstupu:

Pro každé zadání musí program vypsat právě jeden řádek. Pokud není možné rozdělit normohlouček na dvě stejné části se zadaným obvodem, bude na řádku věta "`Rozdeleni neni mozne.`". V ostatních případech vytisknětě větu "`Existuje` *R* `ruznych moznosti.`", kde *R* je počet různých způsobů, jak hlouček rozdělit předepsaným způsobem. Dva způsoby jsou považovány za různé, pokud vzniklé poloviny jsou rozdílné, tedy pokud nelze polovinu z jednoho dělení převést na polovinu podle druhého dělení pomocí operací posunutí, otáčení a zrcadlení.

Příklad vstupu:

```
3
2 5
4 12
6 24
```

Příklad výstupu:

```
Rozdeleni neni mozne.
Existuje 1 ruznych moznosti.
Existuje 10 ruznych moznosti.
```



Praľská spojka

spojka.c, spojka.p, spojka.C
spojka.in, spojka.out,

Dobrá informovanost o aktuální situaci je pro Policii velmi důležitá. V případě poruchy telekomunikačního vedení nemají mezi sebou jednotlivá stanoviště spojení, které tak musí být nahrazeno pomocí poslů. Každý posel může vyřídit vzkaz na daném místě a v daný okamžik. Přestože poslové používají moderních dopravních prostředků (bicyklů), je přeci jen při jejich práci velice důležité znát vzdálenost, na kterou je třeba zprávu dopravit. Proto musí posel vždy volit nejkratší cestu, aby dorazil do cíle své cesty co nejdříve. Kromě toho se někdy může stát, že některé veřejné komunikace není možné kvůli občanským nepokojům použít. V krajním případě se dokonce všechna stanoviště rozpadnou na dvě nebo i více skupin, mezi kterými není žádné spojení.

Policejní složky mají velmi propracovaný projekt infrastruktury, je vždy přesně dáno rozmístění stanovišť v terénu a vzdálenosti mezi nimi. Proto je možné zmapovat vzdálenosti mezi jednotlivými stanovišti a rozhodnout, která cesta je pro případného posla nejhorší. Podle ní se potom zjišťuje, jaká aktuálnost krizového zpravodajství může být zaručena.

Specificace vstupu:

První řádek vstupního souboru obsahuje celé kladné číslo *Z*, za kterým následuje postupně *Z* zadání. Každé zadání začíná řádkem obsahujícím dvě čísla oddělená mezerou. Jsou to po řadě počet jednotek (stanovišť) *J*, 2 <= *J* <= 300 a počet existujících spojnic mezi nimi *S*, 0

<= *S* <= *J*.(*J*-1) / 2. Dále následuje *S* řádků, kaľdý z nich obsahuje právě tři čísla oddělená mezerou. První dvě čísla udávají čísla spojených stanovią» (jednotky jsou číslovány od jedné do *J*) a třetí číslo je délka spojnice mezi nimi. Je přitom moľné, aby mezi nimi existovala také kratąí cesta, která vede přes některé jiné stanoviątě. I takovou samozřejmě můľe posel vyuľít. ®ádná dvojice stanovią» se v rámci jednoho zadání nevyskytne dvakrát.

Specifikace výstupu:

Úkolem je najít takovou dvojici stanovią», mezi kterými je maximální vzdálenost v případě pouľití nejkratąí moľné cesty. Cesta můľe vést pouze po zadaných spojnicích. Pro kaľdé zadání vypíąe program větu "`Nejvetsi vzdalenost je` *V*`.`", kde *V* je délka nejdeląí cesty mezi dvěma stanoviąti, mezi kterými nelze nalézt cestu kratąí. Pokud by zadaná situace znamenala, ľe existuje alespoň jedna dvojice stanovią» taková, ľe mezi nimi neexistuje ľádné spojení, vypíąe program místo toho větu "`Bez spojeni neni veleni!`".

Příklad vstupu:

```
2
5 5
1 2 1
2 3 2
3 4 3
4 5 4
5 1 5
5 2
1 2 1
2 3 1
```

Příklad výstupu:

```
Nejvetsi vzdalenost je 7.
Bez spojeni neni veleni!
```



Tokeny

tokeny.c, tokeny.p, tokeny.C
tokeny.in, tokeny.out,

Důleľitou sloľkou přípravy na očekáváné demonstrace je pozorné sledování toho, co aktivisté chystají. V dneąní době jiľ i odpůrci globalizace běľně pouľívají globální Internet k tomu, aby proti globalizaci protestovali. Nepřekvapí, ľe i ostatní skupiny obyvatel, kteří svolávají veřejné shromáľdění, často vytvářejí webovské stránky informující o chystaných aktivitách. Tento způsob je velmi výhodný k oslovení velkého počtu lidí, ale naątěstí zároveň také pomáhá Policii, aby si udělala předběľný obrázek o tom, co se chystá.

Bohuľel je vąak na současném Internetu stránek velmi mnoho, a tak není snadné nalézt takové, na kterých jsou podobné akce svolávány. Proto bylo rozhodnuto, ľe se pro Policii vytvoří speciální program vyuľívající nejnovějąí poznatky z oblasti umělé inteligence. Úkolem tohoto programu bude procházet internetové stránky a automaticky detekovat „podezřelé" texty. Ty jsou potom nahláąeny policistům k přezkoumání. Základní činností programu je rozbor anglického textu a pokus o určení jeho přibliľného obsahu. K tomu je nutná podrobná analýza textu, nejprve lexikální (na úrovni slov a symbolů), poté syntaktická (na úrovni vět). V tomto příkladě budeme uvaľovat pouze první, lexikální část.

Jak jistě víte, lexikální analyzátor čte posloupnost znaků textu a vytváří posloupnost lexikálních elementů (tokenů). Kdyľ lexikální analyzátor přečte např. text `abc+123`, vytvoří (za předpokladu, ľe text je analyzován podle jazyka pro aritmetický výraz) tokeny [`abc`] [+] [`123`] .