56. ročník matematické olympiády na středních školách

---

19. mezinárodní olympiáda v informatice

In: Karel Horák (editor); Martin Mareš (editor); Peter Novotný (editor); Jaromír Šimša (editor); Jaroslav Švrček (editor); Pavel Töpfer (editor): 56. ročník matematické olympiády na středních školách. Zpráva o řešení úloh ze soutěže konané ve školním roce 2006/2007. 48. mezinárodní matematická olympiáda. 19. mezinárodní olympiáda v informatice. (Czech). Praha: Jednota českých matematiků a fyziků, 2008. pp. 170–185.

Persistent URL: http://dml.cz/dmlcz/405142

19. mezinárodní olympiáda v informatice

Ve dnech 15.–22. 8. 2007 se konal v chorvatském hlavním městě Zagreb 19. ročník Mezinárodní olympiády v informatice (IOI 2007 — International Olympiad in Informatics). Soutěže se zúčastnilo 285 studentů ze 77 zemí celého světa. Většina zemí využila možnosti vyslat na IOI maximální povolený počet čtyři soutěžící, z několika nových účastnických zemí přijela reprezentace menší.

Reprezentační družstvo České republiky bylo sestaveno na základě výsledků dosažených soutěžícími v ústředním kole 56. ročníku Matematické olympiády — kategorie P (programování). Naše družstvo pro IOI 2007 mělo následující složení:

Pavel Klavík, absolvent Gymnázia J. Ressla v Chrudimi,
Miroslav Klimoš, student Gymnázia M. Koperníka v Bílovci,
Josef Pihera, absolvent Gymnázia ve Strakonicích,
Roman Smrž, student Gymnázia E. Krásnohorské v Praze.

Vedoucími české delegace byli jmenováni doc. RNDr. *Pavel Töpfer*, CSc., a Bc. *Petr Škoda*, oba z Matematicko-fyzikální fakulty Univerzity Karlovy v Praze. Vedle této oficiální šestičlenné delegace se 19. mezinárodní olympiády v informatice zúčastnil z České republiky ještě Mgr. *Martin Mareš*, rovněž pracovník MFF UK v Praze, a to z titulu své funkce člena Mezinárodního vědeckého výboru IOI.

Naši studenti měli možnost připravit se na soutěž nejen samostatným studiem, ale zúčastnili se i dvou přípravných akcí. V červnu jsme pro vybrané reprezentanty z České republiky, Polska a Slovenska uspořádali na Matematicko-fyzikální fakultě UK v Praze tradiční týdenní společné česko-polsko-slovenské přípravné soustředění před IOI (CPSPC — Czech-Polish-Slovak Preparation Camp). Na začátku července jsme pak využili skutečnosti, že jsme letos byli pořadateli 14. ročníku Středoevropské olympiády v informatice (CEOI — Central European Olympiad in Informatics) a že jsme na ni mohli jakožto pořádající země pozvat dvě soutěžní družstva. Vedle obvyklé účasti našeho „druhého“ reprezentačního družstva složeného z vybraných mladších studentů (nematurantů) jsme proto na CEOI 2007 do Brna pozvali i naše hlavní reprezentační družstvo připravující se v té době na účast v IOI 2007.

Hlavním pořadatelem letošní mezinárodní olympiády v informatice byla Chorvatská informatická společnost, záštitu nad akcí převzal osobně prezident Chorvatské republiky Mr. Stjepan Mesić. Na uspořádání se finančně podílelo Ministerstvo vědy, školství a sportu Chorvatské republiky a také mnozí sponzoři. Ubytování a stravování účastníků bylo zajištěno v prostorách studentských kolejí Univerzity v Zagrebu, na stejném místě probíhala také všechna jednání mezinárodní jury, vedoucích delegací a rovněž překlady soutěžních úloh do národních jazyků. Prostory pro vlastní soutěž, pro slavnostní zahájení a zakončení akce poskytl magistrát města Zagreb v halách místního veletržního areálu.

Vlastní soutěž byla jako obvykle rozdělena do dvou soutěžních dnů, v každém z nich studenti řešili tři soutěžní úlohy a na práci měli vymezen čas 5 hodin. Každý soutěžící pracuje na přiděleném osobním počítači s nainstalovaným soutěžním prostředím, které umožňuje vyvíjet a testovat programy a odesílat je k vyhodnocení. Výsledné programy jsou testovány pomocí připravené sady testovacích dat a se stanovenými časovými limity. Tím je zajištěna nejen kontrola správnosti výsledků, ale pomocí časových limitů se také odliší kvalita použitého algoritmu. Při testování každé úlohy se používají sady testovacích dat různé velikosti, takže teoreticky správné řešení založené na neefektivním algoritmu zvládne dokončit výpočet pouze pro některé, menší testy. Takové řešení je potom ohodnoceno částečným počtem bodů.

Pořadatelé olympiády letos připravili soutěžní úlohy velmi pečlivě, takže celý proces přípravy úloh, upřesnění jejich formulací a překladu zadání úloh do národních jazyků večer před soutěží probíhal velmi rychle. Soutěžící měli jen minimum dotazů k zadání úloh a neobjevily se ani žádné oprávněné protesty proti hodnocení. Soutěžní úlohy byly dobře navrženy, byly věcně zajímavé, jejich obtížnost byla přiměřená úrovni této soutěže.

Pro všechny účastníky IOI připravují pořadatelé vždy také doprovodný program. Na IOI v Chorvatsku jsme se zúčastnili celodenního výletu do národního parku Plitvická jezera, vedle toho se všem nabízela možnost ve volných chvílích si prohlédnout město Zagreb, navštěvovat místní sportoviště a podnikat turistické výlety do blízkých hor v okolí města.

Večer před odjezdem se konalo slavnostní zakončení soutěže spojené s vyhlášením výsledků. Každá ze soutěžních úloh byla hodnocena maximálně 100 body, takže celkově bylo teoreticky možné získat až 600 bodů. To se ovšem vzhledem k náročnosti úloh nikomu nepodařilo, celkový vítěz dosáhl výsledku 574 bodů. Podle pravidel IOI obdrží na závěr soutěže lepší polovina účastníků některou z medailí, přičemž zlaté, stříbrné a bronzové medaile se udělují přibližně v poměru 1 : 2 : 3 (pochopitelně s ohledem na to, aby soutěžící se stejným bodovým ziskem získali stejnou medaili). Letos bylo uděleno 25 zlatých, 48 stříbrných a 69 bronzových medailí.

Reprezentanti České republiky byli na letošní olympiádě velmi úspěšní, získali jsme tři stříbrné a jednu bronzovou medaili. Výsledky našich soutěžících:

42.	Josef Pihera	333 bodů	stříbrná medaile
53.	Miroslav Klimoš	316 bodů	stříbrná medaile
58.	Roman Smrž	310 bodů	stříbrná medaile
77.	Pavel Klavík	277 bodů	bronzová medaile

Mezinárodní olympiáda v informatice je soutěží jednotlivců, žádné oficiální pořadí zemí se neurčuje. Z hlediska počtu a kvality získaných medailí bychom se v hodnocení národních delegací umístili přibližně kolem pěkného 15. místa z celkového počtu 77 zúčastněných zemí. Patříme tak do nejlepší čtvrtiny zemí, jejichž všichni reprezentanti obdrželi některou z medailí. Nejúspěšnějšími zeměmi letos byla Čína (4 zlaté medaile) a Rusko (3 zlaté a 1 stříbrná). Celkovým vítězem se stal reprezentant Polska, druhé a třetí místo obsadili Číňané. Soutěžícím ze Slovenska se tentokrát vedlo o něco hůře než nám, získali v soutěži dvě stříbrné a jednu bronzovou medaili.

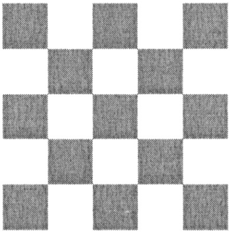
Veškeré informace o soutěži, texty soutěžních úloh i podrobné výsledky všech účastníků lze nalézt na Internetu na adrese

`http://www.hsin.hr/ioi2007/`.

Následující, v pořadí 20. ročník IOI se bude konat v srpnu 2008 v Egyptě. Tamní organizátoři již nyní předběžně pozvali všechny země zúčastněné na IOI v Chorvatsku k účasti v příštím ročníku soutěže. Byla již stanovena i místa konání několika dalších ročníků IOI: 2009 Bulharsko, 2010 Kanada, 2011 Thajsko.

1. Alieni

Roman je bohatý farmář a jeho políčko trávy (ano, není to jen obyčejná tráva) se rozrostlo do slušných rozměrů. Tím se stalo příležitostí pro Romanova kamaráda Myrega, aby předvedl své znalosti mimozemských civilizací a jejich vzorů ze slehlé trávy. Poslední dobou ho zaujaly krásy Chorvatska, a tak se rozhodl, že vytvoří v poli vzor podobný výsostnému znaku Chorvatska, což je šachovnice 5×5 se 13 červenými a 12 bílými poli (obr. 64).



Obr. 64. Šachovnice, která je částí chorvatského výsostného znaku.

Romanovo pole je rozděleno na $N\times N$ políček. Políčko v levém dolním rohu označíme souřadnicemi $(1,1)$ a políčko v pravém horním rohu má souřadnice (N,N).

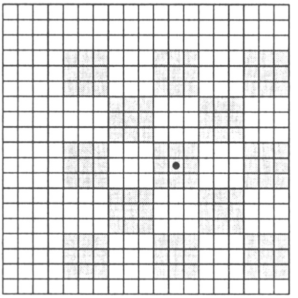
Myreg se rozhodl, že zválí pouze trávu na červených čtvercích šachovnice a zbytek trávy nechá netknutý (přeci jen, kdyby se o tom Roman dozvěděl...). Protože je ale Romanovo pole hodně veliké a Myreg chtěl, aby jeho vzor byl vidět, zvolil si *liché přirozené číslo* $M \geqq 3$ a zválel trávu tak, že jeden čtverec šachovnice odpovídá ploše $M\times M$ políček v Romanově poli. Celá šachovnice se přitom vejde do pole (obr. 65).

Jakmile však Myreg dokončil svoji práci (dělal ji samozřejmě za tmy), překvapil ho náhlý hlas.

„Co tu hledáš, Myregu!“ ozval se policista Pepa, kterého Myreg dobře znal.

„Nic, jen se procházím,“ bránil se Myreg, ale Pepa mu vůbec nevěřil. „Jestli zase ničíš Romanovu úrodu svými hloupými vzory, tak uvidíš!“

„To jsem nebyl já, to byli ALIENI,“ snažil se Myreg zachránit, co se dalo. Pepa se už ale dal na prohlídku pole. Bohužel jeho baterka mu moc dlouho nevydrží a tak ji může zapnout vždy jen na chvíli, aby zjistil, jestli *políčko, na kterém se právě nachází, je slehlé nebo ne.*



Obr. 65. Příklad Romanova pole a Myregova vzoru pro $N = 19$ a $M = 3$. Slehlá políčka jsou označena šedě. Střed vzoru má souřadnice $(12, 9)$ a je označen černou tečkou.

Pepa našel *jedno políčko se slehlou trávou* a potřebuje najít *prostřední políčko* celého Myregova obrazce, aby dokázal, že je to jeho výtvor. Neví ale, jakou zvolil Myreg *hodnotu M velikosti čtverce šachovnice.*

Úloha: Napište program, který pro danou velikost N Romanova pole ($15 \leqq N \leqq 2\,000\,000\,000$), souřadnice jednoho políčka slehlé trávy (X_0, Y_0) a s využitím možnosti zjistit stav trávy na políčku najde souřadnice prostředního políčka Myregova obrazce.

Pepova baterka ale může být použita maximálně 300krát v jednom běhu programu.

Interakce: Tato úloha je interaktivní. Váš program posílá na standardní výstup příkazy, kde má Pepa rozsvítit baterku, a čte odpovědi ze standardního vstupu.

- ▷ Na začátku program načte ze standardního vstupu tři celá čísla N, X_0 a Y_0 oddělená jednou mezerou. Číslo N je velikost Romanova pole a (X_0, Y_0) jsou souřadnice jednoho ze slehlých políček.
- ▷ Pokud chcete prozkoumat pomocí baterky políčko na souřadnicích (X, Y), vypište na standardní výstup řádek ve tvaru „`examine X Y`“. Pokud nejsou souřadnice (X, Y) uvnitř Romanova pole (podmínky $1 \leqq X \leqq N$ a $1 \leqq Y \leqq N$ nejsou splněny) nebo pokud použijete baterku více než 300krát, váš program dostane 0 bodů za tento testovací vstup.
- ▷ Po použití baterky obdržíte na standardním vstupu jeden řádek obsahující slovo „`true`“, pokud je tráva na políčku (X, Y) slehlá, jinak obdržíte řádek se slovem „`false`“.
- ▷ Po nalezení prostředního políčka vypište na standardní výstup řádek

ve tvaru „`solution` $\mathtt{X_C}$ $\mathtt{Y_C}$“, kde (X_C, Y_C) jsou souřadnice prostředního políčka. Běh vašeho programu bude poté automaticky ukončen.

Aby interakce probíhala správně, program musí *nechat vypsat data z bufferu na standardní výstup (flush)* po každém zápisu na standardní výstup; vzorový kód pro každý jazyk je k dispozici.

Vzorové kódy: Vzorové kódy pro všechy tři programovací jazyky jsou k dispozici ke stažení na stránce „Tasks“ soutěžního prostředí. Účelem vzorového kódu je ukázat, jak probíhá interakce; není to správné řešení úlohy.

Hodnocení: V testovacích datech odpovídajících ohodnocení 40 body bude hodnota M nejvýše 100.

Pro každý testovací vstup existuje jediné správné řešení nezávislé na tom, jaké otázky pokládá váš program.

Příklad: V následujícím příkladu jsou příkazy zapsány v levém sloupci, na každém řádku jeden. Odpovědi jsou zapsány v pravém sloupci příslušného řádku.

výstup (příkaz)	*vstup (odpověď)*
	`19 7 4`
`examine 11 2`	`True`
`examine 2 5`	`False`
`examine 9 14`	`False`
`examine 18 3`	`True`
`solution 12 9`	

Testování: V průběhu soutěže máte tři možnosti, jak otestovat svoje řešení.

První možností je simulovat odpovědi manuálně.

Druhou možností je napsat si program, který simuluje odpovědi. Abyste připojili své řešení úlohy k tomuto programu, použijte program „`connect`“, který je k dispozici ke stažení v soutěžním prostředí. Program se používá příkazem „`./connect./solution./device`“ z konzole (místo „solution“ a „device“ uveďte jména vašich programů). Další parametry příkazu budou předány dále simulačnímu programu.

Třetí možností je použít funkci TEST ve vyhodnocovacím systému, kde si můžete otestovat svoje řešení na libovolných testovacích datech. Při použití této funkce je velikost N omezena hodnotou 100.

Testovací data musí obsahovat tyto tři řádky:

- ▷ První řádek obsahuje velikost Romanova pole N a velikost čtverce v šachovnici M;
- ▷ Druhý řádek obsahuje souřadnice X_0 a Y_0 jednoho slehlého políčka, které budou předány vašemu programu;
- ▷ Třetí řádek obsahuje souřadnice X_C a Y_C středu šachovnice.

Vyhodnocovací systém vám poskytne podrobný popis výpočtu zahrnující chybové zprávy, pokud:

- ▷ N nevyhovuje omezením;
- ▷ M není liché přirozené číslo rovné nebo větší než 3;
- ▷ Obrazec se nevejde do Romanova pole;
- ▷ Tráva na políčku (X_0, Y_0) není slehlá.

Následuje příklad správných vstupních dat pro testovací funkci. Příklad odpovídá obr. 65.

```
19 3
 7 4
12 9
```

Korektní vstup pro testovací funkci.

2. Plachty

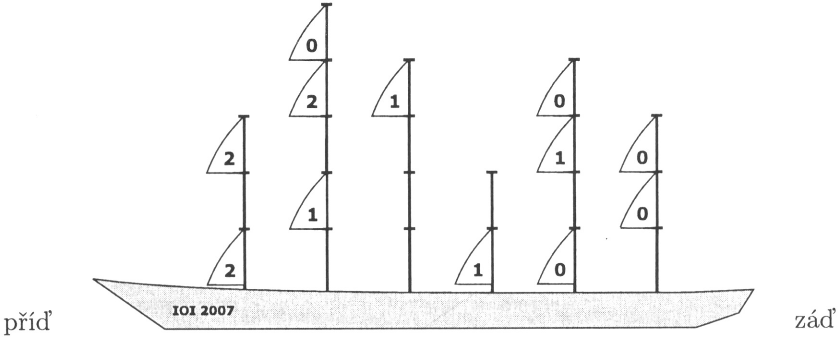
Pirát Pavel si staví novou loď. Loď má N stěžňů rozdělených na úseky jednotkové délky — výška stěžně je rovna počtu jeho úseků. Každý stěžeň je vybaven několika plachtami, každá plachta je umístěna v právě jednom úseku stěžně. Plachty mohou být na stěžni rozmístěny libovolně, ale v každém úseku může být nejvýše jedna z nich.

Různá rozmístění plachet mají ve větru různou účinnost. K plachtám umístěným před jinými plachtami ve stejné výšce se dostává méně větru a jsou proto méně účinné. Pro každou plachtu definujeme její *ztráty* jako celkový počet plachet umístěných na stěžních *ve stejné výšce za* touto plachtou. Pojmy „před“ a „za“ jsou vztaženy k orientaci lodi: na obr. 66 „před“ znamená vlevo a „za“ znamená vpravo.

Celkové ztráty rozmístění plachet jsou rovny součtu ztrát všech plachet na lodi.

Úloha: Napište program, který dostane výšky jednotlivých stěžňů a počty plachet na každém z nich a určí *nejmenší* možné celkové ztráty rozmístění těchto plachet.

Vstup: První řádek vstupu obsahuje celé číslo N ($2 \leqq N \leqq 100\,000$, počet stěžňů na lodi. Každý z následujících N řádků obsahuje dvě celá čísla H a K ($1 \leqq H \leqq 100\,000$, $1 \leqq K \leqq H$), výšku a počet plachet na odpovídajícím stěžni. Stěžně jsou uvedeny v pořadí od přídě k zádi.



Obr. 66. Loď má 6 stěžňů, jejich výšky jsou 3, 5, 4, 2, 4 a 3 odpředu (zleva na obrázku) dozadu. Rozmístění plachet má celkové ztráty 10. Ztráty jednotlivých plachet jsou uvedeny uvnitř plachet.

Výstup: Na výstupu je jedno celé číslo určující minimální možné celkové ztráty plachet.

Poznámka. Při výpočtu používejte 64bitový celočíselný typ (`long long` v C/C++, int64 v Pascalu).

Hodnocení: V testovacích datech odpovídajících ohodnocení 25 body bude celkový možný počet rozmístění plachet roven nejvýše 1 000 000.

Příklad: *vstup*

```
6
3 2
5 3
4 1
2 1
4 3
3 2
```

výstup

```
10
```

Tento příklad odpovídá obr. 66.

3. Povodeň

V roce 1964 zasáhla město Záhřeb katastrofická povodeň. Mnoho budov bylo zcela zničeno, když voda strhla jejich zdi. V této úloze dostanete zjednodušený model města před povodní a máte určit, které zdi zůstanou stát po povodni.

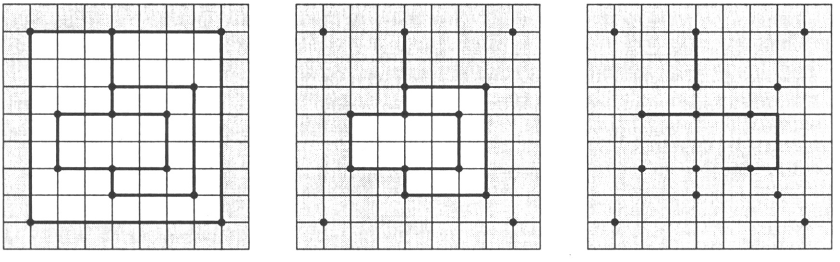
Model je tvořen N body v rovině a W zdmi. *Každá zeď spojuje dvojici zadaných bodů a neprochází žádným jiným bodem.* Model má následující vlastnosti:

▷ Žádné dvě zdi se neprotínají ani nepřekrývají, mohou se však dotýkat svými koncovými body.

▷ Každá zeď je rovnoběžná buď se svislou, nebo s vodorovnou souřadnicovou osou.

Na začátku je všude sucho. V čase nula začne voda zaplavovat vnější oblast (tzn. prostor, který není uvnitř zdí). Přesně po hodině se pod tlakem vody zboří každá zeď, která měla na jedné své straně vodu a na druhé vzduch. Voda pak zaplaví další oblast, jež není zcela ohraničena zdmi. Tím se mohou objevit nové zdi s vodou na jedné a vzduchem na druhé straně. Po další hodině tyto zdi také spadnou a záplava pokračuje dále. Tento proces se opakuje tak dlouho, dokud voda nezaplaví celé město.

Následující obr. 67 ukazuje příklad průběhu záplavy.



Stav v čase nula. Šedé čtverečky představují zaplavenou oblast, bílé suchou (vzduch).

Stav po jedné hodině.

Stav po dvou hodinách. Voda zaplavila celé město a zůstaly stát 4 zdi.

Obr. 67

Úloha: Napište program, který dostane souřadnice N bodů a popis W zdí spojujících tyto body, a určí, které zdi zůstanou stát po povodni.

Vstup: První řádek vstupu obsahuje celé číslo N ($2 \leqq N \leqq 100\,000$), počet bodů v rovině.

Každý z následujících N řádků obsahuje dvě celá čísla X, Y (obě mezi 0 a 1 000 000, včetně), souřadnice jednotlivých bodů. Body jsou očíslovány od 1 do N v pořadí, ve kterém jsou zadány na vstupu. Žádné dva body neleží na stejných souřadnicích.

Další řádek vstupu obsahuje celé číslo W ($1 \leqq W \leqq 2N$), počet zdí.

Každý z následujících W řádků obsahuje dvě různá celá čísla A, B ($1 \leqq A \leqq N, 1 \leqq B \leqq N$) určující, že před povodní stála ve městě zeď spojující body A a B. Zdi jsou očíslovány od 1 do W v pořadí, ve kterém jsou zadány na vstupu.

Výstup: První řádek výstupu bude obsahovat jedno celé číslo K, počet zdí, které zůstanou stát po povodni. Na následujících K řádcích budou zapsány indexy těchto zdí, každý na samostatném řádku. Na jejich pořadí nezáleží.

Hodnocení: V testovacích datech odpovídajících ohodnocení 40 body budou souřadnice všech bodů v rozmezí od 0 do 500.
V těchto testovacích datech a v testovacích datech ohodnocených dalšími 15 body bude N nejvýše 500.

Podrobné informace o výsledcích testování

Během soutěže si můžete zvolit nejvýše 10 odevzdaných řešení této úlohy, která pak budou (co možná nejdříve) vyhodonocena na části oficiálních testovacích dat. Po vyhodnocení budete mít k dispozici přehled výsledků tohoto testování ve vašem soutěžním prostředí.

Příklad:

vstup	*výstup*
15	4
1 1	6
8 1	15
4 2	16
7 2	17
2 3	
4 3	
6 3	
2 5	
4 5	
6 5	
4 6	
7 6	
1 8	
4 8	
8 8	
17	
1 2	
2 15	

```
15 14
14 13
13 1
14 11
11 12
12 4
4 3
3 6
6 5
5 8
8 9
9 11
9 10
10 7
7 6
```

Tento příklad odpovídá situaci na obr. 67.

4. Horníci

Ve dvou dolech pracují skupiny horníků. Těžba uhlí je velmi namáhavá, a tak horníci potřebují stále jíst (podobně jako programátoři). Pokaždé, když do dolu přivezou nějaké jídlo, horníci vytěží jisté množství uhlí. Horníkům vozí tři druhy jídla: maso, fazole a brambory. Při každé dodávce však přivezou pouze jeden druh jídla.

Horníci (stejně jako programátoři) mají rádi pestrou stravu. Čím je jejich strava rozmanitější, tím více pracují (narozdíl od programátorů). Přesněji řečeno, jakmile dorazí nová dodávka jídla, množství vytěženého uhlí závisí na *této dodávce jídla a na dvou předchozích* (nebo na méně, pokud ještě tolik dodávek nedostali):

- ▷ Pokud byly všechny dodávky stejného druhu, horníci následně vytěží jeden vozík uhlí.
- ▷ Pokud se v uvažovaných dodávkách vyskytovaly dva druhy jídla, horníci následně vytěží dva vozíky uhlí.
- ▷ Pokud obsahovala každá dodávka jiný druh jídla, horníci následně vytěží tři vozíky uhlí.

Předem známe druhy všech dodávek jídla a pořadí, v němž budou odeslány do dolů. O každé dodávce jídla můžeme rozhodnout, do kterého dolu bude doručena. Tím můžeme ovlivnit celkové množství vytěženého uhlí. Jednotlivé dodávky jídla není možné dělit, každá musí být doručena vcelku do jednoho nebo druhého dolu.

Oba doly nemusí dostat stejný počet dodávek jídla (dokonce je dovoleno poslat všechno jídlo do jednoho dolu).

Úloha: Dostanete druhy dodávaného jídla v pořadí, v jakém budou do dolů poslány. Napište program, který určí *co největší množství uhlí*, které může být při vhodném rozdělení dodávek jídla mezi oba doly dohromady vytěženo.

Vstup: První řádek vstupu obsahuje jedno celé číslo N ($1 \leqq N \leqq 100\,000$) — počet dodávek jídla.

Druhý řádek je tvořen řetězcem N znaků, představujících druhy jednotlivých dodávek v pořadí, v jakém byly do dolů odeslány. Každý znak bude jedním z velkých písmen »`M`« (pro maso), »`F`« (pro fazole) nebo »`B`« (pro brambory).

Výstup: Na výstupu bude jedno celé číslo — největší možné celkové množství uhlí, které může být vytěženo.

Hodnocení: V testovacích datech odpovídajících ohodnocení 45 body bude počet dodávek N nejvýše 20.

Podrobné informace o výsledcích testování

Během soutěže si můžete zvolit nejvýše 10 odevzdaných řešení této úlohy, která pak budou (co možná nejdříve) vyhodonocena na části oficiálních testovacích dat. Po vyhodnocení budete mít k dispozici přehled výsledků tohoto testování ve vašem soutěžním prostředí.

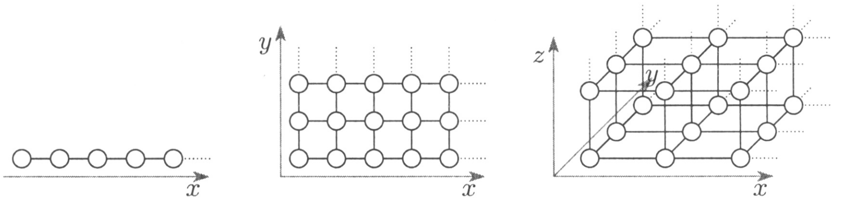
Příklady:

vstup	*vstup*
`6`	`16`
`MBMFFB`	`MMBMBBBBMMMMMBMB`
výstup	*výstup*
`12`	`29`

V levém příkladu můžeme rozdělit dodávky jídla do dolů v pořadí: důl 1, důl 1, důl 2, důl 2, důl 1, důl 2. V dolech budou postupně vytěženy 1, 2, 1, 2, 3 a 3 vozíky uhlí, tzn. celkem 12 vozíků. Tohoto výsledku lze dosáhnout i jinými způsoby.

5. Žvížátka

Pavel a Pepa si hrají s plyšovými zvířátky. Nejprve si zvolí jeden ze tří hracích plánů znázorněných na obr. 68. Každý hrací plán je tvořen políčky (na obrázku jsou zobrazena jako kolečka) uspořádanými do jedno-, dvou- nebo trojrozměrné mřížky.



Hrací plán 1 Hrací plán 2 Hrací plán 3

Obr. 68

Pavel umístí N *zvířátek* na políčka hracího plánu.

Vzdálenost dvou políček je nejmenší počet tahů, které musí zvířátko provést, aby se dostalo z jednoho políčka na druhé. Jedním tahem se může zvířátko přesunout vždy na některé sousední políčko (na obrázku jsou sousední políčka spojena čarou).

Dvě zvířátka si spolu mohou povídat, pokud je vzdálenost jejich políček *nejvýše* D. Úkolem Pepy je spočítat, kolik *párů zvířátek* je umístěno tak, že si zvířátka tvořící pár mohou spolu povídat.

Úloha: Napište program, který dostane typ hracího plánu, umístění zvířátek a číslo D a nalezne požadovaný počet párů.

Vstup: První řádek vstupu obsahuje čtyři celá čísla v tomto pořadí:

- ▷ Typ hracího plánu B ($1 \leqq B \leqq 3$);
- ▷ Počet zvířátek N ($1 \leqq N \leqq 100\,000$);
- ▷ Maximální vzdálenost D, na kterou si zvířátka mohou povídat ($1 \leqq D \leqq 100\,000\,000$);
- ▷ Velikost hracího plánu M (maximální hodnota souřadnice, která se může objevit na vstupu):
  - ⊳⊳ Pro $B = 1$ bude M nejvýše 75 000 000.
  - ⊳⊳ Pro $B = 2$ bude M nejvýše 75 000.
  - ⊳⊳ Pro $B = 3$ bude M nejvýše 75.

Každý z následujících N řádků obsahuje B celých čísel oddělených mezerami — souřadnice jednotlivých zvířátek. Všechny souřadnice budou v rozmezí 1 až M (včetně).

Na jednom políčku může být více zvířátek.

Výstup: Výstup tvoří jedno celé číslo — počet párů zvířátek, která si spolu mohou povídat.

Poznámka. Při výpočtu používejte 64bitový celočíselný typ (`long long` v C/C++, int64 v Pascalu).

Hodnocení: V testovacích datech odpovídajících ohodnocení 30 body bude počet zvířátek N nejvýše 1 000.

Pro každý hrací plán navíc platí, že program, který vyřeší úspešně všechny testovací vstupy na tomto plánu, získá alespoň 30 bodů.

Příklady:

vstup	*vstup*	*vstup*
`1 6 5 100`	`2 5 4 10`	`3 8 10 20`
`25`	`5 2`	`10 10 10`
`50`	`7 2`	`10 10 20`
`50`	`8 4`	`10 20 10`
`10`	`6 5`	`10 20 20`
`20`	`4 4`	`20 10 10`
`23`	*výstup*	`20 10 20`
výstup	`8`	`20 20 10`
`4`		`20 20 20`
		výstup
		`12`

Vysvětlení levého příkladu. Předpokládejme, že zvířátka jsou očíslovanána od 1 do 6 v pořadí, ve kterém jsou uvedena na vstupu. Výsledné čtyři páry jsou:

- ▷ 1-5 (vzdálenost 5)
- ▷ 1-6 (vzdálenost 2)
- ▷ 2-3 (vzdálenost 0)
- ▷ 5-6 (vzdálenost 3)

Vysvětlení prostředního příkladu. Výsledných osm párů je:

- ▷ 1-2 (vzdálenost 2)
- ▷ 1-4 (vzdálenost 4)
- ▷ 1-5 (vzdálenost 3)
- ▷ 2-3 (vzdálenost 3)
- ▷ 2-4 (vzdálenost 4)
- ▷ 3-4 (vzdálenost 3)
- ▷ 3-5 (vzdálenost 4)
- ▷ 4-5 (vzdálenost 3)

6. Trénink

Mirek a Roman se připravují na výroční cyklistický maraton dvojic kolem Chorvatska. Potřebují si proto zvolit tréninkovou trasu.

Máme N měst a M silnic mezi nimi. Každá silnice spojuje dvě města a lze jí projet oběma směry. Přesně $N - 1$ z těchto silnic je *asfaltových*, zatímco ostatní silnice jsou nezpevněné. Silniční síť byla naštěstí navržena tak, že každá dvojice měst je propojena cestou tvořenou pouze asfaltovými silnicemi. Jinými slovy řečeno, *N měst a $N-1$ asfaltových silnic tvoří stromovou strukturu.*

Navíc platí, že každé město je koncovým bodem *nejvýše deseti silnic.*

Tréninková trasa začíná v nějakém městě, prochází některými silnicemi a končí ve stejném městě, kde začala. Mirek a Roman by při tréninku rádi viděli nová místa, a proto zavedli pravidlo, že *žádným městem* ani *žádnou silnicí* neprojedou dvakrát. Tréninková trasa může začínat ve kterémkoliv městě a nemusí procházet všemi městy.

Ve dvojici cyklistů jede vždy jeden vpředu a rozráží tak vzduch tomu druhému. Proto je jízda vpředu těžší a cyklisté se na této pozici pravidelně střídají. Mirek a Roman se chtějí střídat při průjezdu každým městem. Aby byl pro oba trénink stejně náročný, chtějí zvolit trasu tvořenou *sudým počtem silnic.*

Soupeři Mirka a Romana se rozhodli *zablokovat* některé nezpevněné silnice, aby Mirek s Romanem nemohli nalézt *žádnou* tréninkovou trasu podle výše uvedených požadavků. Pro každou nezpevněnou silnici je zadána *cena* (*kladné* celé číslo), která představuje poplatek za zablokování silnice. Je zakázáno blokovat asfaltové silnice.

Úloha: Napište program, který dostane popis sítě měst a silnic a určí *nejnižší možnou celkovou cenu* potřebnou na zablokování silnic takovým způsobem, aby *neexistovala žádná tréninková trasa* vyhovující uvedeným podmínkám.

Vstup: První řádek vstupu obsahuje dvě celá čísla N a M ($2 \leqq N \leqq$ $\leqq 1\,000$, $N \leqq 1 \leqq M \leqq 5\,000$), počet měst a celkový počet silnic.

Na každém z následujících M řádků jsou tři celá čísla A, B a C ($1 \leqq A \leqq N$, $1 \leqq B \leqq N$, $0 \leqq C \leqq 10\,000$), popisující vždy jednu silnici. Čísla A a B jsou různá a určují města spojená touto silnicí. Pokud je C rovno 0, jedná se o asfaltovou silnici, jinak jde o silnici nezpevněnou a C určuje cenu zablokování této silnice.

Každé město je koncovým bodem nejvýše deseti silnic. Žádné dvě silnice nespojují stejnou dvojici měst.

Výstup: Výstupem programu bude jedno celé číslo — nejmenší možná celková cena popsaná v zadání úlohy.

Hodnocení: V testovacích datech odpovídajících ohodnocení 30 body tvoří asfaltové silnice přímou cestu (tj. žádné město není koncovým bodem pro více než dvě asfaltové cesty).

Podrobné informace o výsledcích testování

Během soutěže si můžete zvolit nejvýše 10 odevzdaných řešení této úlohy, která pak budou (co možná nejdříve) vyhodnocena na části oficiálních testovacích dat. Po vyhodnocení budete mít k dispozici přehled výsledků tohoto testování ve vašem soutěžním prostředí.

Příklady:

Vstup

```
5 8
2 1 0
3 2 0
4 3 0
5 4 0
1 3 2
3 5 2
2 4 5
2 5 1
```

výstup

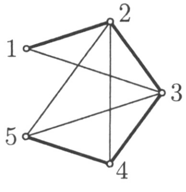
```
5
```

vstup

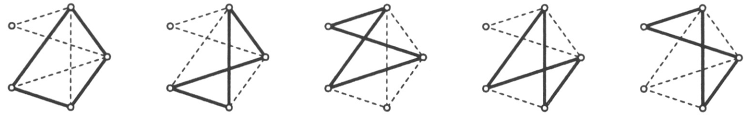
```
9 14
1 2 0
1 3 0
2 3 14
2 6 15
3 4 0
3 5 0
3 6 12
3 7 13
4 6 10
5 6 0
5 7 0
5 8 0
6 9 11
8 9 0
```

výstup

```
48
```



Obr. 69. Silniční síť z prvního příkladu. Asfaltové silnice jsou vyznačeny tučně.



Obr. 70. Mirek a Roman mají pět možných tréninkových tras. Pokud jsou zablokovány silnice 1–3, 3–5 a 2–5, potom nemohou použít ani jednu z těchto pěti tras. Cena zablokování těchto tří silnic je 5. Je také možné zablokovat pouze dvě silnice, 2–4 a 2–5, ale cena je v takovém případě 6, tedy vyšší.