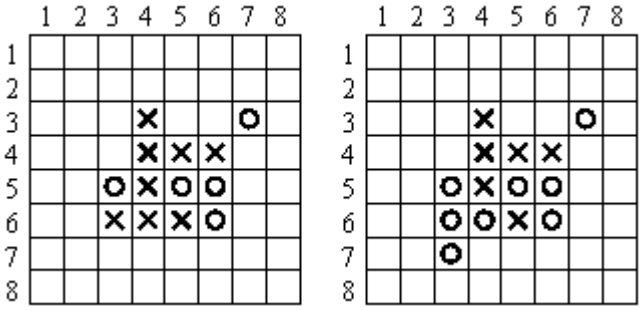
Reversi

(**název programu:** `reversi.c, reversi.p, reversi.C`)

Toníkova dcera Amálka je hravé děvče a protože hraní s panenkami ji už dávno nebaví, hraje s kamarádkami důmyslnou hru *Reversi*. Reversi je hra určená pro dva hráče. Hrací plocha obsahuje 8 x 8 políček, do každého políčka lze podle určitých pravidel umístit nejvýše jeden žeton. Žeton je z jedné strany bílý, z druhé černý. Jeden hráč umísťuje žetony do hrací plochy bílou stranou nahoru, druhý bílou stranou dolu.



Hráč může položit žeton pouze do prázdného políčka a navíc pouze tehdy, "přebarví-li" alespoň jeden žeton protihráče. K přebarvení dojde, jestliže souvislá linie žetonů hráče, který nehrál, se umístěním žetonu stala obklopena žetony protihráče (jeden z obklopujících žetonů musí být ten, který byl právě položen na hrací plochu) a v takové situaci se všechny žetony této linie otočí (tzn. změní barvu). Za souvislou linii se považují sousední žetony umístěné vodorovně, svisle nebo diagonálně.

Při položení žetonu se vždy přebarvuje maximální možný počet žetonů. Obrázek ukazuje příklad, kdy v pozici nalevo byl položen žeton ○ na pozici *[7,3]*. Po přebarvení soupeřových žetonů vznikne situace zobrazená na obrázku vpravo. Hráči se v pokládání žetonů střídají. Nastane-li situace, kdy některý z nich nemůže táhnout tak, aby přebarvil alespoň jeden žeton protihráče, je na tahu znovu druhý hráč. Amálka by ráda měla program, který by mohla využít ve školním turnaji v této hře. Vaším úkolem je napsat tento program.

Specifikace vstupu

Vstup se skládá z *N* zadání. První řádek vstupu obsahuje pouze celé kladné číslo *N*. Dále následují jednotlivá zadání. Každé zadání se skládá z popisu hrací plochy, označení hráče, který je na tahu, a seznamu příkazů, které má program provést.

Hrací plocha je popsána osmi řádky, z nichž každý obsahuje osm znaků. Znak v *i*-tém řádku a v *j*-tém sloupci popisuje stav políčka se souřadnicemi *[i,j]* (řádky i sloupce číslujeme od jedné). K popisu stavu políčka jsou použity tyto znaky:

- '–' znamená prázdné políčko,
- 'X' znamená políčko obsazené hráčem X,
- 'O' znamená políčko obsazené hráčem O.

Jako označení hráče, který je na tahu, je použito písmeno 'X', nebo písmeno 'O'.

Seznam příkazů jsou příkazy uváděné vždy na samostatném řádku a zakončené příkazem 'Q'. Povolené příkazy:

- 'L' znamená požadavek na výpis všech legálních tahů hráče, který je právě na tahu.
- 'M' znamená požadavek na provedení tahu hráčem, který je na tahu. Příkaz 'M' je vždy bezprostředně následován dvojicí číslic *x* a *y*, *1 <= x,y <= 8*. *x* je číslo řádku a *y* je číslo sloupce políčka, na které chce hráč položit svůj žeton. Jestliže hráč, který je na tahu, nemůže táhnout, tak aby přebarvil alespoň jeden žeton soupeře, vztahuje se zadaný tah na druhého hráče.
- 'C' znamená požadavek na výpis počtu žetonů patřících každému z hráčů.
- 'D' znamená požadavek na zobrazení aktuálního obsazení hrací plochy.
- 'Q' ukončuje seznam příkazů.

Specifikace výstupu

Pro každé zadání vypíše program řádek "`Hra c. C:`", kde za *C* se dosadí pořadové číslo zadání. Zadání se číslují vzestupně od jedné. Dále program pro každý zadaný příkaz vypíše odpovídající výstup. Výstupy pro jednotlivé příkazy jsou:

- 'L' - seznam legálních tahů. Každý tah se vypíše jako dvojice po sobě jdoucích číslic *x* a *y*. Dvojice jsou navzájem odděleny čárkou. Všechny tahy se vypíší na jeden řádek. Tahy se vypisují uspořádané vzestupně podle čísla řádku a potom podle čísla sloupce. Pokud neexistuje žádný legální tah, program vypíše řádek "`Zadny legalni tah.`".
- 'M' -- provede požadovaný tah. Pokud hráč, který je právě na tahu, nemůže legálně táhnout na požadované souřadnice, vypíše program "`Neplatny tah.`", jinak tah provede a nevypíše nic. Po provedení tahu se hráči střídají.
- 'C' vypíše počet žetonů hráče X a hráče O ve tvaru "`Hrac O: XX, hrac X: YY`", kde *XX* je počet žetonů hráče O a *YY* je počet žetonů hráče X.
- 'D' zobrazí aktuální stav hrací plochy ve stejném formátu, jako byl použit při zadání počáteční pozice.
- 'Q' vypíše prázdný řádek

Příklad vstupu

```
2
--------
--------
--------
---XO---
---OX---
--------
--------
--------
X
L
M35
C
L
D
Q
XXXXO---
XXXO----
XXO-----
XO------
--------
--------
--------
--------
O
L
M25
C
L
M67
D
Q
```

Příklad výstupu

```
Hra c. 1:
35,46,53,64
Hrac O: 1, hrac X: 4
34,36,56
--------
--------
----X---
---XX---
---OX---
--------
--------
--------

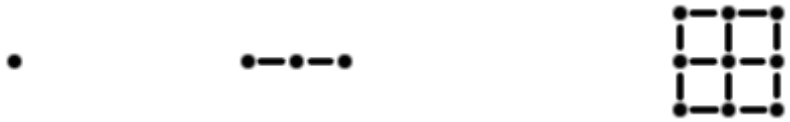
Hra c. 2:
Zadny legalni tah.
Hrac O: 3, hrac X: 12
35
Neplatny tah.
XXXXO---
XXXXX---
XXO-----
XO------
--------
--------
--------
--------
```

Source: https://contest.felk.cvut.cz/98prg/problems.html#reversi

Lodě

(**název programu:** `lode.c, lode.p, lode.C`)

Julie čte zajímavou vědecko-fantastickou knihu, v níž hlavní hrdinka potřebuje vyřešit problém, jak co nejvýhodněji využívat vesmírné nákladní lodě. Lodě mají přepravovat předměty ve tvaru `D`-dimenzionální mřížky, jejichž rozměr v každé dimenzi je 3. V uzlech mřížky jsou koule jednotkové váhy. Mezi uzly vedou spojnice, jejichž hmotnost je vzhledem k hmotnosti koulí zanedbatelná. Váha každého předmětu je tedy rovna počtu jeho uzlů. Cena každého předmětu je rovna součtu počtu uzlů a počtu spojnic. Každá loď má určitou nosnost a my chceme přepravu organizovat tak, aby při každé jízdě měl náklad maximální cenu.



nultá dimenze první dimenze druhá dimenze

Vaším úkolem je napsat program, který pro zadanou nosnost určí, kolik předmětů a jaké dimenze je třeba naložit, aby cena těchto předmětů byla maximální. Předpokládejte, že máte k dispozici neomezený počet kusů od každého předmětu.

Specifikace vstupu

Vstup se skládá z *N* zadání. První řádek vstupu obsahuje pouze celé kladné číslo *N*. Dále následují jednotlivá zadání. Každé zadání je tvořeno jedním řádkem, na němž je jedno celé nezáporné číslo *K*, *K* <= *10000000*, které určuje nosnost lodě.

Specifikace výstupu

Pro každé zadání program vytiskne jeden řádek "$X_m\ X_{m-1}\ \ldots\ X_1\ X_0$", kde $X_m > 0$ a X_i, $0 <= i <= m$ je počet předmětů *i*-té dimenze, které je třeba naložit, aby cena naložených předmětů byla maximální.

Příklad vstupu

```
4
1
100
175
9841
```

Příklad výstupu

```
1
1 0 2 0 1
2 0 1 1 1
1 1 1 1 1 1 1 1 1
```

Source: https://contest.felk.cvut.cz/98prg/problems.html#lode

Výlet

(**název programu:** `vylet.c, vylet.p, vylet.C`)

Julie miluje pěší výlety do hor. Také Toník dříve rád vyrážel za sněhem a sluncem, často vybaven jen ešusem, lžící a spacákem. Jak léta ubíhala, uvykali oba postupně pohodlnému životu ve dvou místnostech svého panelákového bytu. Teď už se jen neradi vzdávají pohodlí, jež nabízí luxusní otoman po babičce či dřevěná lavice z doby husitské. Asi vás proto nepřekvapí, že si chtějí vzít na výlet co možná nejvíce věcí. Zbývá vyřešit otázku, kdo tyto věci ponese. Této otázce věnovali spoustu zimních večerů a po dlouhých diskuzích se dohodli na kompromisu: shromáždí věci, které si chtějí vzít s sebou, u každé věci zjistí její hmotnost, a pak si věci rozdělí v určitém poměru. Vaším úkolem je napsat program, který jim zodpoví otázku, zda lze vybrané věci rozdělit na dvě množiny tak, aby součty hmotností věcí v každé z množin byly přesně v zadaném poměru.

Specifikace vstupu

Vstup se skládá z *N* zadání. První řádek vstupu obsahuje pouze celé kladné číslo *N*. Dále následují jednotlivá zadání. Každé zadání je tvořeno dvěma řádky. Na prvním řádku jsou tři celá kladná čísla *X*, *Y*, *Z*, *1 <= X <= 200*, *1 <= Y,Z <= 100*. Na druhém řádku je *X* celých kladných čísel, která udávají hmotnosti jednotlivých vybraných věcí. Čísla jsou navzájem oddělena mezerou. Žádné z těchto čísel není větší než *500*.

Specifikace výstupu

Pro každé zadání program vytiskne jeden řádek. Pokud lze vybrané věci rozdělit na dvě množiny tak, aby poměr součtů hmotností v každé z množin byl *Y:Z*, bude výstup "`Ano.`", v opačném případě bude výstup "`Ne.`".

Příklad vstupu

```
3
10 1 1
1 1 1 1 1 1 2 2 2 2
5 2 2
3 3 3 3 3
6 2 5
1 2 3 4 5 6
```

Příklad výstupu

```
Ano.
Ne.
Ano.
```

Source: https://contest.felk.cvut.cz/98prg/problems.html#vylet

Tchyně

(**název programu:** `tchyne.c, tchyne.p, tchyne.C`)

Toník je vášnivý cyklista a často podniká výlety do míst blízkých i vzdálenějších. V poslední době však již neusedá na kolo s takovou chutí a nadšením jako dříve. Příčinou této změny je jeho tchyně, která se rozhodla, že jej bude na každé jeho cyklistické vyjížďce doprovázet. Umíte si asi představit tu radost a nadšení, jež se zrcadlily v Tondových očích poté, co mu tchyně sdělila svoje rozhodnutí. Naštěstí mu můžete pomoci. Po několika výletech totiž Tonda zjistil, že tchyně je po nastoupání určitého počtu metrů tak vyčerpaná, že není schopna pokračovat v jízdě. Proto zvolil tuto organizaci společné vyjížďky: on i tchyně vyjedou ze stejného bodu a každý sám se dopraví na předem určené místo srazu. Vaším úkolem je napsat program, který Tondovi pomůže určit, zda tchyně může dojet na místo srazu či nikoliv.

Specifikace vstupu

Vstup se skládá z N zadání. První řádek vstupu obsahuje pouze celé kladné číslo N. Dále následují jednotlivá zadání. Na prvním řádku každého zadání jsou rozměry obdélníkové mapy R a C, $1 <= R,C <= 100$. Následuje R řádků, z nichž každý obsahuje C čísel. Číslo na i-té pozici v j-tém řádku udává nadmořskou výšku bodu o souřadnicích $[i,j]$. Souřadnice levého horního rohu mapy jsou $[1,1]$. Body jsou spojeny cestami, které tvoří pravoúhlou mřížku. Cesta mezi dvěma sousedními body vede vždy pouze do kopce, nebo pouze z kopce, nebo po rovině.

První řádek, který následuje po definici mapy, obsahuje pouze číslo T, které udává počet metrů, jež je tchyně schopna maximálně nastoupat. Další řádek obsahuje pouze čtyři čísla R_1, C_1, R_2, C_2. Čísla jsou navzájem oddělená mezerou. R_1, C_1 jsou souřadnice výchozího bodu a R_2, C_2 jsou souřadnice místa srazu.

Specifikace výstupu

Pro každý dotaz program vytiskne jeden řádek textu. Pokud tchyně může dojet na místo srazu, vytiskne "`Tchyne muze dojet.`", v opačném případě vytiskne "`Tchyne nedojede.`". Předpokládejte, že tchyně jezdí pouze po cestách.

Příklad vstupu

```
2
3 3
1 2 3
2 3 2
3 2 1
1
1 1 3 3
6 5
1 2 3 4 5
9 8 7 6 5
1 2 3 4 5
1 9 9 9 9
7 5 5 5 5
1 1 1 1 1
8
1 1 6 2
```

Příklad výstupu

```
Tchyne nedojede.
Tchyne muze dojet.
```

Source: https://contest.felk.cvut.cz/98prg/problems.html#tchyne

Zlomky

(**název programu:** `zlomky.c, zlomky.p, zlomky.C`)

Malý Toník Okurka je ve věku, kdy děti ještě pravidelně navštěvují základní školu. Většina z nich sice už dávno není přesvědčena o prospěšnosti takového počínání, nicméně z různých důvodů do školy alespoň občas docházejí. Vždyť uznejte, že např. kouřit někde sám daleko za městem je nesrovnatelné s tím, když si zapálíte cigaretu za přítomnosti spolužáků na školním záchodku.

S malým Toníčkem má však paní učitelka jiné trápení. Docházení do školy mu sice nečiní velké potíže, horší je to ale s tím, jak jej ve škole udržet v klidu. Protože je to neposeda, vymyslela si na něj paní učitelka tuto lest: každé ráno dostane Toník několik příkladů na počítání se zlomky a musí zůstat ve škole, dokud je všechny nevyřeší. To se Toníkovi moc nelíbí, protože spíše než zlomky jej zajímá poslední verze *Kernel Hacker's Guide* a nejnovější číslo Playboye. Proto by rád měl program, který za něj příklady vyřeší. Vaším úkolem je tento program napsat.

Specifikace vstupu

Vstup se skládá z *N* zadání. První řádek vstupu obsahuje pouze celé kladné číslo *N*. Dále následují jednotlivá zadání, každé na jednom řádku. Každé zadání se skládá z výrazu, kde výraz je

- *výraz*,
- *výraz* operátor *výraz* (operátor je z množiny {'+', '−', '*', '/'}),
- *číslo* (v desítkové soustavě, číslo může být uvedeno se znaménkem).

Výrazy se vyhodnocují zleva doprava a uplatňují se standardní priority operátorů. Po provedení každé dílčí operace je možné mezivýsledek reprezentovat jako podíl dvou celých čísel, z nichž se obě vejdou do standardního typu *integer* v Pascalu a *int* v Céčku.

Specifikace výstupu

Pro každé zadání program vytiskne jeden řádek "`Vysledek je X.`", kde za *X* se dosadí hodnota zadaného výrazu. Pokud je výsledek ve formě zlomku, musí být maximálně zkrácen (tj. čitatel i jmenovatel musí být nesoudělná čísla).

Příklad vstupu

```
7
9/19-4/19+2/19
2/3+1/3
+21/+20*-2/+3
+4/+5++5/+6
(3/4)/(5/7)
45/4/6/5/7
(10000/10000)*(10000/10000)*(10000/10000)
```

Příklad výstupu

```
Vysledek je 7/19.
Vysledek je 1.
Vysledek je -7/10.
Vysledek je 49/30.
Vysledek je 21/20.
Vysledek je 3/56.
Vysledek je 1.
```

Source: https://contest.felk.cvut.cz/98prg/problems.html#zlomky

Noty

(**název programu:** `noty.c, noty.p, noty.C`)

Toník není žádný velký muzikant. Má však hudbu rád, a proto se rozhodl, že se v ní zdokonalí. Za tímto účelem opouští pravidelně každý čtvrtek svoji krásnou ženu Julii a odchází do jedné z pražských garáží, aby zde vyluzoval zvuky ne zcela nepodobné hudbě. Jeho oblíbeným nástrojem je klavír. Protože ještě nezvládá akordy, doprovází zbytek orchestru alespoň samostatnými tóny. To, jaké tóny má Tonda hrát, se dozví vždy při zkoušce od kapelníka. Tondova paměť však trpí častými návštěvami restaurací 5. cenové skupiny, a tak si Tonda musí jednotlivé tóny zapisovat, aby je do příští zkoušky nezapomněl. Tóny zapisuje jako řetězce složené ze znaků `a`, `b`, `c`, `d`, `e`, `f`, `g`, `h`. I přes Tondovu snahu se často stane, že mu v zápise nějaké tóny chybějí, či naopak přebývají. Nedávno při prohlídce osobních věcí zjistil, že vlastní dva takovéto tónové zápisy. Protože si není jistý tím, který z nich je ten správný, rozhodl se, že z obou zápisů vyškrtne minimální počet tónů tak, aby oba zápisy byly shodné. Tím dosáhne toho, že v zápise určitě nebudou žádné tóny špatně. Vaším úkolem je napsat program, který pro zadané dva tónové zápisy určí, jaký celkový nejmenší počet tónů je třeba vyškrtnout, abychom dostali dva shodné zápisy.

Specifikace vstupu

Vstup se skládá z *N* zadání. První řádek vstupu obsahuje pouze celé kladné číslo *N*. Dále následují jednotlivá zadání. Každé zadání se skládá ze dvou řádků. Na každém z nich je jeden řetězec, který představuje jeden tónový zápis. Délka každého řetězce je nejméně 1 a nejvýše 1000.

Specifikace výstupu

Pro každé zadání program vytiskne jeden řádek "`Je treba vyskrtnout X tonu.`", kde za *X* se dosadí minimální počet tónů, které je třeba vyškrtnout, aby zbylé řetězce byly shodné.

Příklad vstupu

```
2
abab
bbabc
cdacbed
cdhacb
```

Příklad výstupu

```
Je treba vyskrtnout 3 tonu.
Je treba vyskrtnout 3 tonu.
```

Source: https://contest.felk.cvut.cz/98prg/problems.html#noty

Podnikatel

(**název programu:** `podnikatel.c, podnikatel.p, podnikatel.C`)

Malý Toník plánuje stát se podnikatelem a vydělat spoustu peněz. Protože nerad něco odkládá, začal již svůj sen uskutečňovat. Zatím zkoušel štěstí na střelnici u kolotočů, na hracích automatech a dokonce i při karetní hře Boží požehnání, lidově též Gotes. Bohužel, štěstí mu moc nepřálo, a tak jeho podnikání vykazuje ztrátu. Není ovšem všemu konec, neboť Toník objevil novou hazardní hru Master-Mind, která by mu mohla veškeré ztráty vynahradit.

Dříve než však přistoupí na hru o peníze, chce se Master-Mind důkladně naučit. Za tímto účelem potřebuje program, který dokáže ohodnotit jeden pokus v této hře. Vaším úkolem je napsat tento program. Program dostane na vstupu dvě posloupnosti $A_1, A_2, \ldots A_M$ a $B_1, B_2, \ldots B_M$. Jeho úkolem je určit počet míst, na kterých jsou v obou posloupnostech stejná čísla, a dále počet čísel, která se vyskytují v obou posloupnostech na různých místech.

Nechť Z je množina takových dvojic (I,J), $1 <= I,J <= M$, že $A_I = B_J$ a nechť pro každé dva různé prvky této množiny (I_1,J_1) a (I_2,J_2) platí, že $I_1 <> I_2$ a $J_1 <> J_2$. Potom označíme $X(Z)$ počet takových prvků množiny Z, pro které $I = J$, a $Y(Z)$ počet prvků, pro které $I <> J$.

Nyní definujeme relaci uspořádání nad všemi možnými množinami Z takto: $Z_1 <= Z_2$ právě tehdy, když platí jedna z následujících dvou možností:

- $X(Z_1) < X(Z_2)$
- $X(Z_1) = X(Z_2)$ a zároveň $Y(Z_1) <= Y(Z_2)$

Úkolem programu je zjistit hodnoty $X(Z_{max})$ a $Y(Z_{max})$, kde Z_{max} je největší ve smyslu uvedené relace. Nazvěme $X(Z_{max})$ počet černých kolíčků a $Y(Z_{max})$ počet bílých kolíčků.

Specifikace vstupu

Vstup se skládá z N zadání. První řádek vstupu obsahuje pouze celé kladné číslo N. Dále následují jednotlivá zadání. Každé zadání je tvořeno třemi řádky. Na prvním řádku každého zadání jsou dvě čísla M a K, $1 <= M,K <= 1000000$. Hodnota M udává počet členů posloupnosti, K je maximální číslo, které se může v posloupnostech vyskytnout. Na druhém řádku každého zadání jsou čísla $A_1, A_2 \ldots A_M$, navzájem oddělena mezerami. Na třetím řádku každého zadání jsou čísla $B_1, B_2 \ldots B_M$, také navzájem oddělena mezerami.

Specifikace výstupu

Pro každé zadání program vypíše řádek "`Cernych: X, bilych: Y`", kde za *X* se dosadí počet černých kolíčků a za *Y* se dosadí počet bílých kolíčků.

Příklad vstupu

```
3
5 3
1 2 3 1 2
2 3 1 2 3
1 2
1
2
9 9
1 2 3 4 5 6 7 8 9
2 3 1 4 5 6 8 9 7
```

Příklad výstupu

```
Cernych: 0, bilych: 4
Cernych: 0, bilych: 0
Cernych: 3, bilych: 6
```

Source: https://contest.felk.cvut.cz/98prg/problems.html#podnikatel