

Váha

Amina má šest mincí očíslovaných od **1** do **6**. Ví, že všechny mince mají různou hmotnost, a má je podle jejich hmotnosti uspořádat. Za tímto účelem bude používat speciální váhu.

Tradiční váha má dvě misky a používá se tak, že se na každou misku dá jedna mince a váha určí, která z nich je těžší.

Nová váha, jež Amina má, je složitější. Má čtyři misky označené **A**, **B**, **C** a **D**. Váha má čtyři různé režimy nastavení, kde každé z nich odpoví na jiný dotaz ohledně hmotnosti mincí. Používá se tak, že Amina na každou z misek **A**, **B** a **C** položí právě jednu minci. Ve čtvrtém režimu nastavení musí navíc položit právě jednu minci na misku **D**.

Čtyři režimy nastavení váhy dovolují odpovídat na následující čtyři dotazy stran hmotnosti mincí:

1. Která z mincí na miskách **A**, **B** a **C** je nejtěžší?
2. Která z mincí na miskách **A**, **B** a **C** je nejlehčí?
3. Která z mincí na miskách **A**, **B** a **C** je co do hmotnosti prostřední, tj. není ani nejlehčí ani nejtěžší ze všech tří?
4. Z mincí na miskách **A**, **B** a **C** uvažujme pouze ty, jež jsou těžší než mince na misce **D**. Existují-li takové, která z nich je nejlehší? V opačném případě, tedy neexistují-li žádná taková, pak která z mincí na miskách **A**, **B** a **C** je nejlehčí?

Úloha

Napište program, který Amině uspořádá její mince dle jejich hmotnosti. Program se může dotazovat její váhy a porovnávat tak hmotnost mincí. Váš program dostane několik testovacích případů k řešení, každý sestávající z nové množiny šesti mincí.

Váš program musí implementovat funkce `init` a `orderCoins`. Během každého spuštění programu vyhodnocovač nejprve právě jednou zavolá funkci `init`. Tím vám sdělí počet testovacích případů a umožní inicializovat proměnné. Následně vyhodnocovač jednou pro každý testovací případ zavolá `orderCoins()`.

- `init(T)`
 - `T`: Počet testovacích případů, které váš program bude muset vyřešit během tohoto spuštění. `T` je celé číslo v rozsahu **1, ..., 18**.
 - Funkce nevrací žádnou návratovou hodnotu.
- `orderCoins()`
 - Funkce je volána právě jednou pro každý testovací případ.

- Funkce musí určit správné pořadí mincí pomocí volání funkcí `getLightest()`, `getHeaviest()`, `getMedian()` a/nebo `getNextLightest()` vyhodnocovače.
- Jakmile funkce určí správné pořadí mincí, musí zavolat funkci `answer()` vyhodnocovače.
- Po zavolání `answer()` funkce `orderCoins()` musí skončit (bez návratové hodnoty).

Ve svém programu můžete použít následující funkce vyhodnocovače:

- `answer(C)` — váš program použije tuto funkce ke sdělení, že našel odpověď.
 - `C`: Pole délky 6 obsahující správné pořadí mincí. Hodnoty `C[0]` až `C[5]` představují čísla mincí (tj. hodnoty od **1** do **6**) v pořadí od nejlehčí po nejtěžší.
 - Váš program smí zavolat tuto funkci pouze z funkce `orderCoins()` a to právě jednou.
 - Funkce nevrací žádnou návratovou hodnotu.
- `getHeaviest(A, B, C)`, `getLightest(A, B, C)`, `getMedian(A, B, C)` — odpovídají režimům nastavení 1, 2, resp. 3 na váze.
 - `A, B, C`: Mince položené na miskách **A, B, C**. `A, B` a `C` jsou tři celá čísla, každé v rozsahu **1** až **6** včetně.
 - Každá funkce vrací jedno z čísel `A, B, C`: číslo odpovídající mince. Např. `getHeaviest(A, B, C)` vrátí číslo nejtěžší ze tří daných mincí.
- `getNextLightest(A, B, C, D)` — odpovídá režimu nastavení 4 na váze.
 - `A, B, C, D`: Mince na miskách **A, B, C, D**. `A, B, C, D` musejí být čtyři různá celá čísla v rozsahu **1** a **6** včetně.
 - Funkce vrátí jedno z čísel `A, B, C`: číslo mince určené dle režimu nastavení 4. Tedy vrácená mince je nejlehčí z těch mincí **A, B, C**, které jsou těžší než mince na misce **D**, nebo jestliže žádná z nich není těžší než **D**, jednoduše vrátí nejlehčí z mincí na miskách **A, B, C**.

Bodování

Tato úloha nemá žádné podúlohy, místo toho bude bodování založeno na tom, kolik vážení (celkový počet volání vážících funkcí `getLightest()`, `getHeaviest()`, `getMedian()` a/nebo `getNextLightest()` vyhodnocovače) váš program provede.

Váš program bude spuštěn vícekrát s několika testovacími případy v každém spuštění. Necht' r je počet spuštění vašeho programu. Toto číslo je dáno testovacími daty. Jestliže váš program neuspořádá mince správně v žádném testovacím případě žádného běhu, dostane 0 bodů. V opačném případě bude obodován následovně.

Necht' Q je nejmenší číslo takové, že je možné uspořádat jakoukoli posloupnost šesti mincí s použitím Q vážení. Abychom úlohu ztížili, číslo Q nesdělujeme.

Předpokládejme, že největší počet vážení ze všech testovacích případů všech běhů je $Q + y$ pro nějaké celé číslo y . Uvažme jedno spuštění programu. Necht' největší počet vážení ze všech T testovacích případů v tomto běhu je $Q + x$ pro nějaké nezáporné celé číslo x . (Jestliže použijete pro

každý testovací případ Q vážení, pak $x = 0$.) Potom za toto spuštění obdržíte $\frac{100}{r((x+y)/5+1)}$ bodů, zaokrouhloeno na dvě desetinná místa *dolů*.

Například jestliže váš program použije Q vážení při každém spuštění, pak obdržíte 100 bodů.

Příklad

Předpokládejme, že mince jsou uspořádány **3 4 6 2 1 5** od nejlehčí po nejtěžší.

volání	vrátí	popis
getMedian(4, 5, 6)	6	Mince 6 je prostřední (medián) z mincí 4, 5, 6 .
getHeaviest(3, 1, 2)	1	Mince 1 je nejtěžší z mincí 1, 2 a 3 .
getNextLightest(2, 3, 4, 5)	3	Mince 2, 3, 4 jsou všechny lehčí než 5 , je tudíž vrácena nejlehčí z nich, tedy 3 .
getNextLightest(1, 6, 3, 4)	6	Mince 1 a 6 jsou obě těžší než 4 . Z mincí 1 a 6 je mince 6 nejlehčí.
getHeaviest(3, 5, 6)	5	Mince 5 je nejtěžší z mincí 3, 5, 6 .
getMedian(1, 5, 6)	1	Mince 1 je prostřední (medián) z 1, 5, 6 .
getMedian(2, 4, 6)	6	Mince 6 je prostřední (medián) z 2, 4, 6 .
answer([3, 4, 6, 2, 1, 5])		Program našel pro tento testovací případ správnou odpověď.

Ukázkový vyhodnocovač

Ukázkový vyhodnocovač čte vstup v následujícím formátu:

- řádek **1**: T — počet testovacích případů
- každý z řádků **2 až $T + 1$** : posloupnost **6** různých čísel od **1** do **6**: pořadí mincí od nejlehčí po nejtěžší.

Např. vstup sestávající ze dvou testovacích případů, kdy mince jsou uspořádány **1 2 3 4 5 6** a **3 4 6 2 1 5**, vypadá následovně:

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

Ukázkový vyhodnocovač vypíše pole předané jako parametr funkci `answer()`.