

## 秤 (Scales)

Amina は、**1** から **6** までの番号のついた **6** 枚のコインを持っている。彼女はコインの重さが全て異なることを知っている。彼女はこれらのコインを重さ順に並べたいと思っている。そのために彼女は新しい天秤を開発した。

従来の天秤は **2** つの皿を持っており、それぞれの皿にコインを乗せることでどちらのコインが重いのかを調べることができる。

Amina の開発した新しい天秤はより複雑である。この天秤は **A, B, C, D** のラベルのついた **4** つの皿を持っている。また、**4** つの異なる設定があり、それぞれコインに関する異なる質問の答えを得ることができる。この天秤を使うためには、まず **A, B, C** の皿にちょうど **1** 枚ずつコインを置かなければならない、**4** 番目の設定で天秤を使うためには、さらに **D** の皿にもちょうど **1** 枚のコインを置かなければならない。

**4** つのそれぞれの設定で天秤が答えることができる質問は以下の通りである。

- A, B, C** の皿のコインのうち最も重いコインはどれか？
- A, B, C** の皿のコインのうち最も軽いコインはどれか？
- A, B, C** の皿のコインのうち中間の重さを持つコイン（すなわち、最も重いコインでも最も軽いコインでもないコイン）はどれか？
- A, B, C** の皿のコインのうち、**D** の皿に乗っているコインより重いものだけに注目する。もしそのようなコインが存在する場合は、そのうちで最も軽いコインはどれか？そうでない場合は、**A, B, C** の皿のコインのうち最も軽いコインはどれか？

## 課題 (Task)

Amina の持っている **6** 枚のコインを重さ順に並べるプログラムを作成せよ。プログラムでは Amina の天秤にコインの重さを比べる質問をすることができる。あなたのプログラムにはいくつかのテストケースが与えられる。

あなたのプログラムでは関数 `init` と関数 `orderCoins` が実装されていなければならない。あなたのプログラムが実行されるたびに、採点プログラムは初めに **1** 度だけ関数 `init` を呼び出す。これは、あなたにテストケースの個数を与え、また、変数の初期化ができるようにするものである。その後、採点プログラムは各テストケースにつきちょうど **1** 回ずつ関数 `orderCoins` を呼び出す。

■ `init(T)`

- `T`: **1** 回の実行であなたのプログラムが回答しなければならないテストケースの個数を表す **1** 以上 **18** 以下の整数。
- この関数は戻り値を持たない。

- `orderCoins()`
  - この関数は各テストケースにつきちょうど **1** 回ずつ呼び出される。
  - この関数では、採点プログラムで用意された関数 `getHeaviest`, `getLightest`, `getMedian`, `getNextLightest` を利用し、Amina のコインの正しい順番を決定しなければならない。ただし、利用しない関数があってもよい。
  - 重さの順番が決定できるようになったとき、この関数は採点プログラムで用意された関数 `answer` を呼び出してそれを報告する。
  - 関数 `answer` を呼び出した後は、この関数は `return` しなければならない。また、この関数は戻り値を持たない。

あなたのプログラムでは、採点プログラムで用意された以下の関数を利用することができる。

- `answer(W)` — あなたのプログラムは答えを報告するためにこの関数を利用しなければならない。
  - `W`: コインの正しい順番を表す長さ **6** の配列。 `W[0]` から `W[5]` は重さの軽い順に並んだコインの番号 (つまり、**1** から **6** の番号) でなければならない。
  - あなたのプログラムは、関数 `orderCoins` からこの関数を、各テストケースにつきちょうど **1** 回呼び出さなければならない。
  - この関数は戻り値を持たない。
- `getHeaviest(A, B, C)`, `getLightest(A, B, C)`, `getMedian(A, B, C)` — これらの関数はそれぞれAmina の天秤の設定 **1**, **2**, **3** に対応している。
  - `A`, `B`, `C`: それぞれ、皿 ***A***, ***B***, ***C*** に乗せるコインの番号を表す **1** 以上 **6** 以下の相異なる整数である。
  - 各関数は、各質問の答えに対応する ***A***, ***B***, ***C*** のいずれかの整数を返す。例えば、`getHeaviest(A, B, C)` は **3** 枚のうち最も重いコインの番号を返す。
- `getNextLightest(A, B, C, D)` — この関数はAmina の天秤の設定 **4** に対応している。
  - `A`, `B`, `C`, `D`: それぞれ、皿 ***A***, ***B***, ***C***, ***D*** に乗せるコインの番号を表す **1** 以上 **6** 以下の相異なる整数である。
  - 各関数は、質問の答えに対応する ***A***, ***B***, ***C*** のいずれかの整数を返す。すなわち、***A***, ***B***, ***C*** のうち ***D*** より重いコインのうち最も軽いコインの番号を返す。ただし、***A***, ***B***, ***C*** が全て ***D*** より軽い場合は、***A***, ***B***, ***C*** のうち最も軽いコインの番号を返す。

## 採点方式 (Scoring)

この課題には小課題は存在しない。その代わりに、あなたの得点は天秤の使用回数 (すなわち、関数 `getHeaviest`, `getLightest`, `getMedian`, `getNextLightest` の呼び出し回数の合計) によって評価される。

あなたのプログラムは複数回実行され、各実行には複数個のテストケースが含まれる。あなた

のプログラムの実行回数を  $r$  とする。この値はテストデータに対して固定である。もしあなたのプログラムが正しく答えられなかったテストケースが存在した場合は、 $0$  点となる。そうでない場合、各実行の得点が次のように独立に計算される。

$Q$  を、どのようなコインの並びに対しても  $Q$  回天秤を使用すれば正しい順番が確定できるような最小の数とする。ただし、課題をより面白くするために  $Q$  の値は問題文には提示しない。

全ての実行の全てのテストケースのうちの天秤の使用回数の最大値が  $Q + y$  だったとする。そして、あなたのプログラムのある  $1$  つの実行について考える。この実行の全てのテストケースのうちの天秤の使用回数の最大値が  $Q + x$  だったとする。ただし、全てのテストケースで天秤の使用回数が  $Q$  回以下だった場合は  $x = 0$  とする。このとき、この実行に対する得点は  $\frac{100}{r((x+y)/5+1)}$  の値を小数点以下  $2$  桁になるように切り捨てた値となる。

特に、あなたのプログラムの天秤の使用回数がいずれの実行のいずれのテストケースに対しても  $Q$  以下であった場合、あなたは  $100$  点を得る。

## 例 (Examples)

コインが重さが軽い順に  $3\ 4\ 6\ 2\ 1\ 5$  と並んでいることを考える。

関数呼び出し	戻り値	説明
getMedian(4, 5, 6)	6	コイン $6$ がコイン $4, 5, 6$ のうち中間の重さのコインである。
getHeaviest(3, 1, 2)	1	コイン $1$ がコイン $3, 1, 2$ のうち最も重いコインである。
getNextLightest(2, 3, 4, 5)	3	コイン $2, 3, 4$ はいずれもコイン $5$ より軽いため、 $3$ つのうち最も軽いコイン $3$ が返される。
getNextLightest(1, 6, 3, 4)	6	コイン $1, 6$ がコイン $4$ より重い。コイン $1$ とコイン $6$ ではコイン $6$ の方が軽い。
getHeaviest(3, 5, 6)	5	コイン $5$ がコイン $3, 5, 6$ のうち最も重いコインである。
getMedian(1, 5, 6)	1	コイン $1$ がコイン $1, 5, 6$ のうち中間の重さのコインである。
getMedian(2, 4, 6)	6	コイン $6$ がコイン $2, 4, 6$ のうち中間の重さのコインである。
answer([3, 4, 6, 2, 1, 5])		プログラムはこのテストケースにおける正しい答えを発見した。

## 採点プログラムのサンプル (Sample grader)

採点プログラムのサンプルは以下のフォーマットで入力を読み込む：

- $1$  行目：  $T$  — テストケースの個数。
- $2$  行目から  $T + 1$  行目：  $1$  以上  $6$  以下の  $6$  つの相異なる数の列: 重さの軽い順のコインの番号。

例えば、 $1\ 2\ 3\ 4\ 5\ 6$  の順と  $3\ 4\ 6\ 2\ 1\ 5$  の順の  $2$  つのテストケースからなる場合は以下のようになる。

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

採点プログラムのサンプルは、関数 `answer` の引数の配列を出力する。