

ウォーリーを探せ (Where's Waldo?)

問題名	ウォーリーを探せ (whereswaldo)
時間制限	11 sec
メモリー制限	1 GB

長さ N の隠された順列 P_0, P_1, \dots, P_{N-1} がある。この順列は一様ランダムに生成されたことが保証されている。この順列は $1, 2, 3, \dots, N$ の数字をそれぞれちょうど 1 回含み、ある未知の順番で並んでいる。

あなたは場所 l と r を選び、「 $P_l + P_{l+1} + \dots + P_r$ は何か?」という形式の質問をすることができる。

あなたの仕事は、 P の中の 1 の場所をできるだけ少ない回数の質問で探すことである。スコアは質問の回数によって決められる。

インタラクション

あなたのプログラムはまず 1 行に書かれた 2 つの整数 T と N を読み込む。 T はあなたのプログラムがテストされるラウンドの回数であり、 N は P の長さである。

この後、 T ラウンドが行われる:

ラウンドが始まると、質問をすることができる。 1 行に “? a b” と出力することにより、場所 a と b の間 (それらの場所も含む) の値の和を質問することができる。 ($0 \leq a \leq b \leq N - 1$)。

それぞれの質問の後に、あなたのプログラムはその区間の値の総和を表す 1 つの整数を読み込まなければならない。

あなたが 1 の位置を特定したときには、 i を $P_i = 1$ となるようなインデックスとして、1 行に “! i” の形式で出力せよ。この出力の後には次のラウンドが始まる。

質問をした後には必ず標準出力を flush せよ。さもなければ、あなたのプログラムは Time Limit Exceeded と判定される可能性がある。

Python では、 `print()` で自動的に flush される。C++ では、 `cout << endl;` は新しい行を出力し flush も行う。もし `printf` を使う場合は、 `fflush(stdout)` を用いよ。

制約と評価方法

あなたのプログラムは $N = T = 1000$ を満たす単一のテストケースについてテストされる。各ラウンドの順列はランダムに生成されたことが保証されている。

もしあなたの解法がいずれかのラウンドで間違った推測を行った場合、あなたの提出は *Wrong Answer* と判定される。

そうでない場合、以下の方法でスコアが計算される:

$$\text{score} = \min\left(220 - \frac{M}{2500}, 100\right)$$

ただし、 M はあなたのプログラムが T ラウンドを通して行った質問の回数の総和である。

スコアは最も近い整数に丸められる。もしスコアが負の数になった場合には 0 点として扱われる。

例えば、もし 550 000 回以上の質問を行った場合は 0 点となり、もし 300 000 回もしくはそれ未満の回数の質問を行った場合は 100 点となる。その中間では、スコアは直線的に変化する。

テストのためのツール

あなたが解法のテストをすることを容易にするため、シンプルなツールがダウンロードできるようになっている。Kattis の問題ページの一番下の “attachments” を見よ。このツールは必ずしも使う必要はなく、また変更を加えることも許される。ただし、実際に Kattis 上で採点に用いられるプログラムとこのツールは異なることに注意せよ。

使用例 ($T=1000$, $N=10$ としたとき):

Python でプログラムを書いた場合は、例えば `solution.py` を書いた場合、(通常は `pypy3 solution.py` と実行される):

```
python3 testing_tool.py pypy3 solution.py <<<"1000 10"
```

C++ でプログラムを書いた場合は、まずそれをコンパイルして (例えば、`g++ -std=gnu++17 solution.cpp -o solution.out` と打つなどして) その後以下を実行せよ:

```
python3 testing_tool.py ./solution.out <<<"1000 10"
```

例

サンプルでは $T = 2$, $N = 10$ である。2 つのラウンドのうち 1 つ目のラウンドでは、隠された順列は “6 10 8 7 9 1 2 4 5 3” である。初めの質問? 0 9 はすべての値の総和を聞いており、その値は確かに 55 である。2 つ目の質問? 0 4 に対する答えは $6 + 10 + 8 + 7 + 9 = 40$ である。

採点プログラムの出力	あなたのプログラムの出力
2 10	
	? 0 9
55	
	? 0 4
40	
	? 5 5
1	
	! 5
	? 0 0
1	
	! 0