



一流の団子職人 (Super Dango Maker)

JOI 君は団子職人である。JOI 君の店で用いられる団子にはそれぞれ色が定まっており、色は N 種類ある。色には 1 から N までの番号が付けられている。

JOI 君の店の人気メニューはきれいな串団子である。きれいな串団子とは、色が相異なる合計 N 個の団子に 1 本の串を刺したものである。

JOI 君は色 1 から色 N までの団子をそれぞれ M 個ずつ、合計 $N \times M$ 個の団子を用意している。これらの団子には 1 から $N \times M$ までの番号が付けられている。JOI 君はこれらの団子と M 本の串からきれいな串団子を M 本作りた。

団子の色を間違えてはならないため、JOI 君は団子チェッカーを用いることにした。団子チェッカーにいくつかの団子の番号を指定して質問すると、それらの団子と十分な量の串を用いて最大何本のきれいな串団子を作れるかを知ることができる。

JOI 君は団子チェッカーに何度か質問をすることで、 M 本のきれいな串団子を作るために、 $N \times M$ 個ある団子を N 個ずつ M 組に分けたい。このとき、それぞれの組には色 1 から色 N までの団子が 1 つずつ含まれていなければならない。

JOI 君は団子チェッカーに 50 000 回以下の質問を行うことで、団子を組分けしたい。

団子の情報が与えられたとき、団子チェッカーに 50 000 回以下の質問を行うことで団子を組分けする JOI 君の戦略を実装したプログラムを作成せよ。



実装の詳細

あなたは1つのファイルを提出しなければならない。

あなたの提出するファイルは `dango3.cpp` という名前である。このファイルは以下の関数を実装しなければならない。そのプログラムは `#include` プリプロセッサ指令によって `dango3.h` を読み込むこと。

- `void Solve(int N, int M)`

この関数は各テストケースにおいて1回だけ呼び出される。

- 引数 N は団子の色の種類の数 N である。
- 引数 M は JOI 君が作りたきれいな串団子の本数 M である。

あなたのプログラムは以下の関数を呼び出すことができる。

- ★ `int Query(const std::vector<int> &x)`

この関数を用いて、団子チェッカーに質問を行う。

- 引数 x は、団子チェッカーに質問する団子の番号のリストである。
- 戻り値は、指定した団子と十分な量の串を用いて作れるきれいな串団子の本数の最大値である。
- 引数 x の各要素は、1以上 $N \times M$ 以下でなければならない。これが満たされていない場合、不正解 [1] と判定される。
- 引数 x の各要素はすべて異ならなければならない。これが満たされていない場合、不正解 [2] と判定される。
- 関数 `Query` を 50000 回を超えて呼び出してはならない。50000 回を超えて呼び出した場合、不正解 [3] と判定される。

- ★ `void Answer(const std::vector<int> &a)`

この関数を用いて、きれいな串団子を作るための団子の組を解答する。

- 引数 a は、きれいな串団子を作るための団子の番号のリストである。
- 引数 a の長さは、 N でなければならない。これが満たされていない場合、不正解 [4] と判定される。
- 引数 a の各要素は、1以上 $N \times M$ 以下でなければならない。これが満たされていない場合、不正解 [5] と判定される。
- a の要素に同じ値を累計で2回以上指定してはならない。これが満たされていない場合、不正解 [6] と判定される。
- a で指定された各団子と1本の串を用いてきれいな串団子を作ることができない場合、不正解 [7] と判定される。
- 関数 `Answer` はちょうど M 回呼び出される必要がある。関数 `Solve` の実行の終了時に関数



`Answer` の呼び出し回数が M 回でなかった場合、不正解 [8] と判定される。

重要な注意

- 内部での使用のために他の関数を実装したり、グローバル変数を宣言するのは自由である。
- あなたの提出したプログラムは、標準入力・標準出力、あるいは他のファイルといかなる方法でもやりとりしてはならない。ただし、標準エラー出力にデバッグ情報等を出力することは許される。

コンパイル・実行の方法

作成したプログラムをテストするための、採点プログラムのサンプルが、コンテストサイトからダウンロードできるアーカイブの中に含まれている。このアーカイブには、提出しなければならないファイルのサンプルも含まれている。

採点プログラムのサンプルは1つのファイルからなる。そのファイルは `grader.cpp` である。作成したプログラムをテストするには、これらのファイル `grader.cpp`, `dango3.cpp`, `dango3.h` を同じディレクトリに置き、次のようにコマンドを実行する。

```
g++ -std=gnu++17 -O2 -o grader grader.cpp dango3.cpp
```

コンパイルが成功すれば、`grader` という実行ファイルが生成される。

実際の採点プログラムは、採点プログラムのサンプルとは異なることに注意すること。採点プログラムのサンプルは単一のプロセスとして起動する。このプログラムは、標準入力から入力を読み込み、標準出力に結果を出力する。

採点プログラムのサンプルの入力

採点プログラムのサンプルは標準入力から以下の形式で入力を読み込む。

$$N M$$
$$C_1 C_2 \cdots C_{N \times M}$$

C_i ($1 \leq i \leq N \times M$) は、団子 i の色を表す、1以上 N 以下の整数である。

採点プログラムのサンプルの出力

採点プログラムのサンプルは標準出力へ以下の情報を出力する (引用符は実際には出力されない)。



- 正解の場合、関数 `Query` の呼び出し回数が “Accepted: 2022” のよう出力される。
- 不正解の場合、不正解の種類が “Wrong Answer [4]” のよう出力される。

実行するプログラムが複数の不正解の条件を満たした場合、表示される不正解の種類はそれらのうち 1 つのみである。

制約

すべての入力データは以下の条件を満たす。 C の意味については「採点プログラムのサンプルの入力」の項目を参照せよ。

- $1 \leq C_i \leq N$ ($1 \leq i \leq N \times M$).
- 各 j ($1 \leq j \leq N$) に対し、 $C_i = j$ となる i ($1 \leq i \leq N \times M$) がちょうど M 個存在する。
- N, M は整数である。
- C_i ($1 \leq i \leq N \times M$) は整数である。

小課題

1. (2 点) $N = 4$, $M = 4$.
2. (5 点) $N = 100$, $M = 10$.
3. (15 点) $N = 200$, $M = 25$.
4. (78 点) $N = 400$, $M = 25$.



やりとりの例

採点プログラムのサンプルが読み込む入力の例と、それに対応する関数の呼び出しの例を以下に示す。

入力例 1	関数の呼び出しの例		
	呼び出し	呼び出し	戻り値
3 2	Solve(3, 2)		
3 3 1 2 1 2		Query([])	0
		Query([4, 2, 1, 3])	1
		Query([3, 4, 5])	0
		Query([2, 6, 5])	1
		Query([6, 5, 4, 3, 2, 1])	2
		Answer([1, 6, 5])	
		Answer([2, 3, 4])	

この入力例はどの小課題の制約も満たさないことに注意すること。

コンテストサイトからダウンロードできるファイルのうち、`sample-02.txt` は小課題 1 の制約を満たす。