



コイン (coins)

ジャムシード (Jamshid) の敵であるザッハーク (Zahhak) は、ジャムシードの娘のアルナヴァーズ (Arnavaz) とシャハルナズ (Shahrnaz) を捕らえた。しかし、ザッハークは彼女たちに、パズルを解くことで自由の身になるチャンスを与えることにした。

ザッハークは 8×8 のチェス盤を持っている。それぞれのセルには、下図のように、0 から 63 までの番号によるラベルが付けられている。彼は、64 個のセルそれぞれに、1 枚ずつコインを置いた。ラベル c のセルには、特別なコインが置かれている。そのコインは、物理的には他のコインと同一だが、呪われている。それぞれのコインは、表または裏が見えるように置かれている。

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

ザッハークは姉妹を夕食に招き、パズルについて説明した。夕食後は、彼女たちは別々の部屋に行かなければならない。その後、ザッハークはアルナヴァーズの部屋に行き、チェス盤を見せて c の値 (呪われたコインの置かれたセルのラベル) を伝える。アルナヴァーズはコインの位置を変更することはできないが、1 枚以上、 k 枚以下のコインを裏返す (表と裏をひっくり返す) ことができる。同じコインを複数回裏返してもよい。次に、ザッハークはもう一方の部屋に行き、シャハルナズにチェス盤を見せる。もし、シャハルナズが呪われたコインの位置を特定することができれば、2 人の姉妹は解放される。姉妹は、夕食の最中に戦略について話し合うことができるが、夕食後は連絡を取ることはできない。

あなたの課題は、姉妹がザッハークのパズルを解くのを助けることである。



実装の詳細 (Implementation details)

2つの異なるプロシージャを実装せよ:

```
int[] coin_flips(int[] b, int c)
```

- このプロシージャには、アルナヴァーズの戦略を実装せよ。
- b : 長さ 64 の整数配列であり、ザッハークがアルナヴァーズに見せるチェス盤を表す。 $b[i]$ ($0 \leq i \leq 63$) の値は 0 または 1 である。それぞれ、セル i に置かれたコインが、表が見えるように置かれているか、または、裏が見えるように置かれていることを表す。
- c : 呪われたコインの置かれたセルのラベル
- このプロシージャは、アルナヴァーズがコインを裏返すセルのラベルを含む配列を返すこと。配列の長さは 1 以上、 k 以下である。この配列には、同じ値が複数回含まれてもよい。

```
int find_coin(int[] b)
```

- このプロシージャには、シャハルナーズの戦略を実装せよ。
- b : 長さ 64 の整数配列であり、ザッハークがシャハルナーズに見せるチェス盤 (アルナヴァーズが何枚かのコインを裏返した後のもの) を表す。
- このプロシージャは、呪われたコインの位置 c を返すこと。

シナリオは全部で T 個ある。それぞれのシナリオにおいて、採点用プログラムは、プロシージャ `coin_flips` を呼び出す。その戻り値に応じて、採点用プログラムは、チェス盤を更新してプロシージャ `find_coin` を呼び出す。採点システムにおいて、これらのプロシージャは別々のプログラムとして呼び出されることに注意せよ。最初のプログラムにおいて、プロシージャ `coin_flips` は、各シナリオごとに 1 回ずつ呼び出される。プロシージャ `find_coin` の呼び出しは、2 つ目のプログラムが行う。2 つのプログラムにおけるシナリオの呼び出し順序は同一とは限らないので、それぞれのシナリオにおけるあなたの実装の振る舞いは、シナリオの順序によらないものでなければならない。

制約 (Constraints)

- $1 \leq T \leq 1000$
- $0 \leq c \leq 63$



小課題 (Subtasks)

1. (10 点) $c < 2, k = 1$
2. (15 点) $c < 3, k = 1$
3. (10 点) $k = 64$
4. (15 点) $k = 8$
5. (50 点) $k = 1$

例 (Example)

姉妹は次のように決めたと仮定する: アルナヴァーズは、呪われたコインのみを裏返す。シャハルナーズは、裏が見えるように置かれたコインのうちの 1 枚の位置を返す (そのようなコインが存在しない場合は、0 を返す)。明らかに、これは単なる例であって、正しい戦略ではない。

採点プログラムは、次のようにプロシージャを呼び出す:

```
coin_flips([0,0,...,0,0], 63)
```

この例において、 b は 0 のみからなる長さ 64 の配列であり、チェス盤の全てのコインが表が見えるように置かれていることを表す。このプロシージャは、一つの値のみからなる長さ 1 の整数配列 [63] を返す。

次に、採点プログラムは、番号 63 のセルに置かれたコインを裏返して、次のようにプロシージャを呼び出す:

```
find_coin([0,0,...,0,1])
```

このプロシージャは 63 を返す。これは、呪われたコインの正しい位置である。

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、以下の形式で入力を読み込む:

- 1 行目: T
- ブロック i ($0 \leq i \leq T - 1$): シナリオ i を表す 9 行からなるブロック
 - 1 行目: c
 - $2 + j$ 行目 ($0 \leq j \leq 7$): テーブル b の j 行目を表す長さ 8 のバイナリ (0/1) 文字列

採点プログラムのサンプルは、以下の形式で出力する:

- $1 + i$ 行目 ($0 \leq i \leq T - 1$): シナリオ i に対するあなたの解答の判定結果