



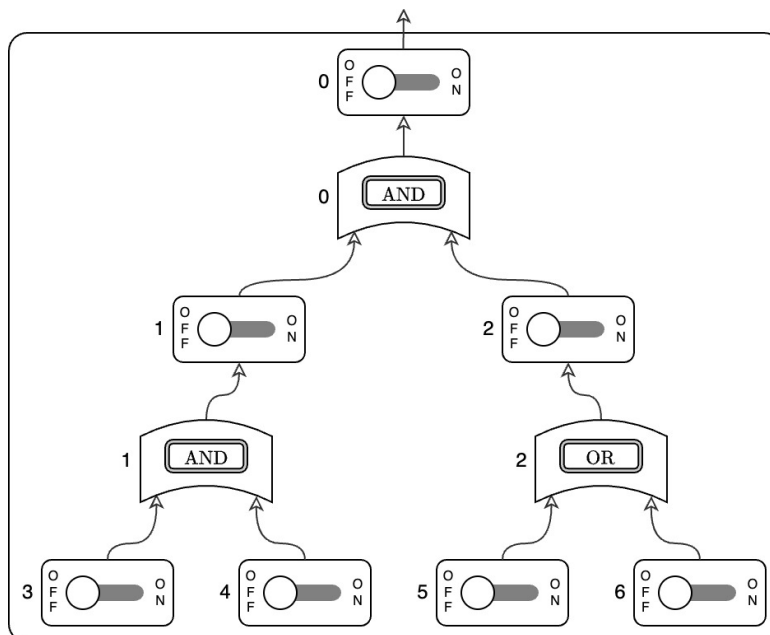
電子回路 2 (Circuit 2)

JOI 君は電子回路セットで遊んでいる。電子回路セットは N 個の **AND** 部品, N 個の **OR** 部品 と, 1 個の回路ボードからなる。回路ボードは $2N + 1$ 個のスイッチと N 個の部品スロットで構成され, 各部品スロットに **AND** 部品か **OR** 部品を設置することで使用することができる。回路ボードは, 設置された部品とスイッチの状態に応じて, 以下のように 0 または 1 の値を出力する。

回路ボードの仕様

- スイッチには 0 から $2N$ までの番号が付けられており, 各スイッチは **ON** または **OFF** の状態を持つ。各スイッチは下に書かれたように 0 または 1 の値を出力する。
- 部品スロットには 0 から $N - 1$ までの番号が付けられている。各部品スロットは下に書かれたように 0 または 1 の値を出力する。
- 各スイッチおよび各部品スロットの出力は, 以下の規則によって番号の大きなものから順（同じ番号のスイッチと部品スロットについては部品スロットが先）に定まる。
 - $j = 2N, 2N - 1, \dots, N$ について, スイッチ j は以下を出力する。
 - スイッチ j が **OFF** であるとき, 0 を出力する。
 - スイッチ j が **ON** であるとき, 1 を出力する。
 - $j = N - 1, N - 2, \dots, 0$ について, 部品スロット j の出力を x としたとき, スイッチ j は以下を出力する。
 - スイッチ j が **OFF** であるとき, x を出力する。
 - スイッチ j が **ON** であるとき, $1 - x$ を出力する。
 - $i = N - 1, N - 2, \dots, 0$ について, 部品スロット i は 2 個のスイッチ U_i, V_i と接続している。ここで, U_i, V_i は $i < U_i < V_i \leq 2N$ を満たす。スイッチ U_i の出力を x , スイッチ V_i の出力を y としたとき, 部品スロット i は以下を出力する。
 - 部品スロット i に **AND** 部品が設置されているとき, $\min(x, y)$ を出力する。
 - 部品スロット i に **OR** 部品が設置されているとき, $\max(x, y)$ を出力する。
- $j = 1, 2, \dots, 2N$ について, $U_i = j$ または $V_i = j$ であるような i ($0 \leq i \leq N - 1$) がただ 1 つ存在する。
- 回路ボードの出力は, スイッチ 0 の出力と等しい。

例えば, $N = 3, U_0 = 1, V_0 = 2, U_1 = 3, V_1 = 4, U_2 = 5, V_2 = 6$ で, 部品スロット 0, 1, 2 にそれぞれ **AND** 部品, **AND** 部品, **OR** 部品が設置されている場合, 回路ボードは以下の図のように表される。



さて、JOI 君がすべての部品スロットに AND 部品を設置しようとしたところ、設置した部品の中に 最大 R 個の OR 部品がまぎれ込んでいることが明らかになった。AND 部品と OR 部品は見た目で区別することができないため、回路ボードを使用して AND 部品と OR 部品を区別しなければならない。あなたの仕事は、JOI 君に以下の形式の質問を 1000 回まで行い、どの部品スロットに OR 部品が設置されているかを特定することである。

- あなたは JOI 君に $2N + 1$ 個のスイッチをどのような状態にするか指示する。JOI 君は指示された通りにスイッチの状態を変更し、回路ボードの出力をあなたに教える。

回路ボードの回路の接続の情報と設置されている OR 部品の個数の上限が与えられたとき、1000 回以下の質問で、どの部品スロットに OR 部品が設置されているかを特定するプログラムを作成せよ。



入出力

この問題は **インタラクティブ問題**（あなたが作成したプログラムと採点プログラムが標準入出力を介して対話を行う形式の問題）である。

最初に、回路ボードの回路の接続の情報と設置されている OR 部品の個数の上限が以下の形式で標準入力から与えられる。

$$\begin{array}{l} NR \\ U_0 \ V_0 \\ U_1 \ V_1 \\ \vdots \\ U_{N-1} \ V_{N-1} \end{array}$$

あなたのプログラムはこれを受け取った後、採点プログラムとやり取りを行わなければならない。

JOI 君に質問をするには、以下の形式で採点プログラムとやり取りせよ。

- まず、以下の形式で標準出力に出力せよ。

? s

- s は ‘0’ と ‘1’ からなる長さ $2N + 1$ の文字列であり、 s の $j + 1$ 文字目 ($0 \leq j \leq 2N$) が ‘0’ ならばスイッチ j を OFF にすることを、 s の $j + 1$ 文字目が ‘1’ ならばスイッチ j を ON にすることを表す。
 - 文字列 s の長さが $2N + 1$ でない場合、不正解 [1] と判定される。
 - 文字列 s に ‘0’ でも ‘1’ でもない文字が含まれる場合、不正解 [2] と判定される。
- 次に、質問に対する回答 x が以下の形式で標準入力から与えられる。

x

ここで、 x は $0, 1, -1$ のいずれかであり、それぞれ以下を表す。

- x が 0 である場合、JOI 君は s にしたがってスイッチの状態を変更し、変更後の回路ボードの出力が 0 であったことを表す。
- x が 1 である場合、JOI 君は s にしたがってスイッチの状態を変更し、変更後の回路ボードの出力が 1 であったことを表す。
- x が -1 である場合、採点プログラムがあなたのプログラムを不正解であると判定したことを表す。この場合、ただちにあなたのプログラムを終了せよ。

質問を 1000 回より多く行ってはならない。1000 回より多く質問した場合、不正解 [3] と判定される。



どの部品スロットに OR 部品が設置されているかを解答するには、以下の形式で標準出力に出力せよ。

! t

- t は ‘&’ と ‘|’ からなる長さ N の文字列であり、 t の $i+1$ 文字目 ($0 \leq i \leq N-1$) が ‘&’ であるとき、部品スロット i に AND 部品が設置されていることを、 t の $i+1$ 文字目が ‘|’ であるとき、部品スロット i に OR 部品が設置されていることを表す。
- 文字列 t の長さが N でない場合、不正解 [4] と判定される。
- 文字列 t に ‘&’ でも ‘|’ でもない文字が含まれる場合、不正解 [5] と判定される。
- 実際に各部品スロットに設置されている部品の種類が文字列 t の表すものと異なる場合、不正解 [6] と判定される。
- 解答はちょうど 1 回行わなければならない。2 回以上解答した場合、不正解 [7] と判定される。あなたのプログラムの終了時に解答が 1 回も行われていなかった場合、不正解 [8] と判定される。

上に挙げたいずれでもない形式で標準出力に出力した場合、不正解 [9] と判定される。

重要な注意

- 各出力の最後には、必ず標準出力を flush せよ。flush しない場合、実行時間制限超過と判定される可能性がある。
 - C++ のプログラムを提出する場合、`cout << endl;` によって、改行されるとともに自動的に flush される。もし `printf` を使用する場合、`fflush(stdout)` を用いよ。
 - Python のプログラムを提出する場合、入力に `input()` を使う限り自動的に flush される。
- 質問の際に -1 を受け取った場合、ただちにプログラムを終了せよ。終了しなかった場合、実行結果は不定である。

採点に関する注意

実際の採点プログラムは適応的 (adaptive) ではなく、やりとりの初めから固定された答えを持つ。



テストツール

作成したプログラムをテストするためのテストツール `testing_tool.py` 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。このツールを必ずしも使う必要はなく、また変更することも許される。実際の採点プログラムはテストツールとは異なることに注意すること。

あなたのプログラムが C++ で実装されている場合、作成したプログラムをテストするには、以下のような手順に従えばよい。

- まず、あなたのプログラムをコンパイルして実行ファイルを生成せよ。例えば、あなたのプログラムが `circuit.cpp` という名前の場合、以下のようなコマンドを実行すれば `circuit` という実行ファイルが生成される。

```
g++ -std=gnu++20 -O2 -o circuit circuit.cpp
```

- 次に、以下のコマンドを実行せよ。

```
python3 testing_tool.py ./circuit
```

あなたのプログラムが Python で実装されている場合、例えば、あなたのプログラムが `circuit.py` という名前なら、以下のコマンドを実行せよ。

```
python3 testing_tool.py pypy3 circuit.py
```

テストツールの入力

T をあなたのプログラムが解答すべき長さ N の文字列とする。テストツールは標準入力から以下の形式で入力を読み込む。

```
N R
U0 V0
U1 V1
⋮
UN-1 VN-1
T
```



テストツールの出力

テストツー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, 質問回数が “Accepted: 22” のように出力される.
- 不正解の場合, 不正解の種類が “Wrong Answer [4]” のように出力される.

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

- $1 \leq N \leq 8\,000$.
- $1 \leq R \leq \min(N, 120)$.
- $i < U_i < V_i \leq 2N$ ($0 \leq i \leq N-1$).
- $j = 1, 2, \dots, 2N$ について, $U_i = j$ または $V_i = j$ であるような i ($0 \leq i \leq N-1$) がただ 1 つ存在する.
- N, R, U_i, V_i は整数である ($0 \leq i \leq N-1$).

小課題

1. (1 点) $N = 1$.
2. (8 点) $N \leq 1\,000$, $R = 1$.
3. (10 点) $N \leq 1\,000$.
4. (21 点) $U_i = i + 1$, $V_i = N + 1 + i$ ($0 \leq i \leq N-1$), $R \leq 70$.
5. (9 点) $U_i = i + 1$, $V_i = N + 1 + i$ ($0 \leq i \leq N-1$).
6. (25 点) $U_i = 2i + 1$, $V_i = 2i + 2$ ($0 \leq i \leq N-1$), $R \leq 70$.
7. (9 点) $U_i = 2i + 1$, $V_i = 2i + 2$ ($0 \leq i \leq N-1$).
8. (14 点) $R \leq 70$.
9. (3 点) 追加の制約はない.



やりとりの例

テストツールが読み込む入力の例と、それに対応するやりとりの例を以下に示す.

入力例 1	採点プログラムの出力	あなたのプログラムの出力
1 1	1 1	
1 2	1 2	
		? 010
	1	
		? 011
	1	
		? 111
	0	
		!

1 回目の質問では, 回路ボードの出力は以下のように計算できる.

- スイッチ 1,2 の状態はそれぞれ ON, OFF であるから, スイッチ 1,2 はそれぞれ 1,0 を出力する.
- 部品スロット 0 には OR 部品が設置されていて, 接続するスイッチ 1,2 はそれぞれ 1,0 を出力しているから, 部品スロット 0 は $\max(1,0) = 1$ を出力する.
- スイッチ 0 の状態は OFF で, 部品スロット 0 の出力は 1 であるから, スイッチ 0 の出力は 1 である.
- したがって, 回路ボードの出力は 1 である.

この入力例はすべての小課題の制約を満たす.



The 5th Japanese Olympiad in Informatics for Girls (JOIG 2024/2025)
Spring Training/Qualifying Trial
March 20–24, 2025 (Komaba, Tokyo)

Contest 3 – Circuit 2

入力例 2	採点プログラムの出力	あなたのプログラムの出力
3 3	3 3	
1 2	1 2	
3 4	3 4	
5 6	5 6	
&&		? 0001001
	0	
		? 0001110
	1	
		? 0000011
	0	
		! &&

問題文中の図がこの入力例の回路ボードを表している。

この入力例は小課題 3, 6, 7, 8, 9 の制約を満たす。

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-01-in.txt は入力例 1 に対応し、sample-02-in.txt は入力例 2 に対応する。また、sample-03-in.txt は小課題 3, 4, 5, 8, 9 の制約を満たし、sample-04-in.txt は小課題 3, 6, 7, 8, 9 の制約を満たす。