お絵かきパズル (Paint By Numbers)

*お絵かきパズル*は有名なパズルゲームである．ここでは，このゲームの単純な 1 次元の場合を考える．このゲームでは，プレイヤーに n 個のマスからなる行が与えられる．これらのマスは左から右に $0, 1, \ldots, n-1$ の番号がつけられている．プレイヤーはそれぞれのマスを黒色または白色に塗らなければならない．ここでは，'X'で黒色のマスを，'_'で白色のマスを表す．

プレイヤーにはヒントと呼ばれる，k 個の正の整数からなる数列 $c = [c_0, \ldots, c_{k-1}]$ が与えられる．プレイヤーは連続する黒色のマスからなるブロックがちょうど k 個できるように行を塗らなければならない．さらに，左から i 番目のブロック（最も左のブロックを 0 番目とする）にはちょうど c_i 個の黒色のマスが含まれていなければならない．たとえば，ヒントが $c = [3, 4]$ である場合，パズルの解答では連続する黒色のマスからなるブロックがちょうど 2 つできていなければならず，それらのうち片方の長さは 3 で，もう片方の長さは 4 でなければならない．したがって，$n = 10$ かつ $c = [3, 4]$ のとき，"_XXX__XXXX"という解答はヒントの条件を満たす．"XXXX_XXX__"という解答は，黒色のマスのブロックが正しい順序に並んでいないので，ヒントの条件を満たさないことに注意せよ．また，"__XXXXXXX_"という解答も，黒色のマスが 2 つの別のブロックではなく，1 つのブロックとなっているので，ヒントの条件を満たさない．

あなたには部分的に解かれたお絵かきパズルの問題が与えられる．すなわち，あなたには n, c の値，および黒色に塗られなければならないマスと白色に塗られなければならないマスの情報が与えられる．あなたの課題はマスの色に関する追加の情報を得ることである．

パズルの*正しい解答*とは，パズルの解答であって，ヒントの条件を満たし，色が分かっているマスは与えられた色に塗られているものをいう．あなたのプログラムは，すべての正しい解答において黒色に塗られているマスと，すべての正しい解答において白色に塗られているマスを見つけなければならない．与えられる入力では少なくとも一つ正しい解答があると仮定してよい．

実装の詳細 (Implementation details)

あなたは以下の関数を実装しなければならない．

- `string solve_puzzle(string s, int[] c)`
  - `s`: 長さ n の文字列．各 i（$0 \leq i \leq n-1$）について，`s` の i 番目の文字は以下のとおりである．
    - マス i が黒色に塗られなければならないとき，'X'．
    - マス i が白色に塗られなければならないとき，'_'．
    - マス i の色に関する情報がないとき，'.'
  - `c`: 上で定義された，ヒントを表す長さ k の数列．
  - この関数は長さ n の文字列を返さなければならない．さらに，その文字列の i

番目（$0 \leqq i \leqq n-1$）の文字は，以下のとおりでなければならない．
    - マス i がすべての正しい解答において黒色に塗られているとき，'X'．
    - マス i がすべての正しい解答において白色に塗られているとき，'_'．
    - 上記以外のとき（すなわち，2 つの正しい解答が存在して，片方ではマス i が黒色に，もう片方では白色に塗られているとき），'?'．

C 言語の場合，引数および戻り値の形式が少し異なる．
- `void solve_puzzle(int n, char* s, int k, int* c, char* result)`
    - `n`: 文字列 `s` の長さ（マスの個数）．
    - `k`: 数列 `c` の長さ（ヒントの数）．
    - 他の引数は上記のものと同じである．
    - 長さ n の文字列を返す代わりに，この関数は，引数として渡された文字列 `result` に答えを書き込まなければならない．

この問題で使われる文字の ASCII コードは以下のとおりである．
- 'X' : 88,
- '_' : 95,
- '.' : 46,
- '?' : 63.

実装の詳細については，各言語のテンプレートファイルを参照せよ．

例 (Examples)

例 1 (Example 1)

```
solve_puzzle("..........", [3, 4])
```

このパズルに対するすべての正しい解答を以下に示す．
- "XXX_XXXX__"
- "XXX__XXXX_"
- "XXX___XXXX"
- "_XXX_XXXX_"
- "_XXX__XXXX"
- "__XXX_XXXX"

これらを観察すると，マス 2，6，7 はいずれの解答においても黒色に塗られることが分かる．また，それ以外のマスは黒色に塗られることも塗られないこともある．したがって，この例の答えは "??X???XX??" となる．

例 2 (Example 2)

```
solve_puzzle("........", [3, 4])
```

この例のパズルの正しい解答は一意に定まり，この例の答えは "XXX_XXXX" となる．

例 3 (Example 3)

```
solve_puzzle("..._._....", [3])
```

この例では，マス 4 は必ず白色に塗られることが分かる．なぜなら，マス 3 とマス 5 の白色

のマスの間に長さ 3 のブロックを配置することが出来ないからである．したがって，この例の答えは “???___????” となる．

例 4 (Example 4)

```
solve_puzzle(".X.......", [3])
```

この例の入力に沿ったパズルの正しい解答は，以下に示す 2 通りしかない．

- “XXX_______”，
- “_XXX______”．

したがって，この例の答えは “?XX?______” となる．

小課題 (Subtasks)

全ての小課題で，$1 \leqq k \leqq n$ かつ $1 \leqq c_i \leqq n$（$0 \leqq i \leqq k-1$）を満たす．

1. (7 点): s は ‘.’ のみからなり，$n \leqq 20$ かつ $k = 1$ を満たす．
2. (3 点): s は ‘.’ のみからなり，$n \leqq 20$ を満たす．
3. (22 点): s は ‘.’ のみからなり，$n \leqq 100$ を満たす．
4. (27 点): s は ‘.’ と ‘_’ のみからなり，$n \leqq 100$ を満たす．
5. (21 点): $n \leqq 100$ を満たす．
6. (10 点): $n \leqq 5\,000$ かつ $k \leqq 100$ を満たす．
7. (10 点): $n \leqq 200\,000$ かつ $k \leqq 100$ を満たす．

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: 文字列 s．
- 2 行目: 整数 k と，それに続く k 個の整数 $c_0, \ldots, c_{k-1}$．