



Paint By Numbers 按數目填色遊戲

按數目填色是個有名的智力遊戲。我們現在考慮的是這個遊戲的一維版本。在這個遊戲中，玩家會收到一個含有 n 個格位的行格。這些格位由左至右分別編號為 0 至 $n - 1$ 。玩家必須要為這些格位填上黑色或白色。我們用 'X' 代表填上黑色的格位並用 '_' 來代表填上白色的格位。

系統將會給予玩家一個提示，這個提示是一個含有 k 個正整數的序列 $c = [c_0, \dots, c_{k-1}]$ 。而玩家必須要根據提示為格位填色，填色的要求是在這行上，填上黑色的格位必須組成 k 個連續格的區間。而在由左開始的第 i (由 0 開始) 個區間上必須含有 c_i 個黑色格位。例如若提示為 $c = [3, 4]$ ，則這遊戲的正確填色方法是必須含有兩個由連續黑色格位組成的區間，這兩個區間的長度分別是 3 和 4 。因此若 $n = 10$ 且 $c = [3, 4]$ 的話，則其中一個可能的正確填色方法是 "_XXX_XXXX"。要注意的是 "XXXX_XXX_" 並不是一個正確的填色方法，因為黑色區間的長度順序不正確。同時 "__XXXXXXXX_" 亦不是一個正確的填色方法，因為它只含有一個區間而非兩個分開的區間。

系統將給予你一個已填有部份解的遊戲行格。換句話說，系統將給予你 n 及 c 的值，並且亦會給予你部份已填上黑色或白色的格位資料。你的工作是從中推算出更詳細的填色資料作為答案。

更加詳細來說，一個正確的遊戲解是一個填色方法，這個方法合符提示給出的條件，亦合符已知已填有黑色或白色格位的要求。你的程式必須找出在所有的遊戲解中都會填上黑色的格位及都是填上白色的格位。

你可以假設系統給你的提示是必定可以找到最少一個可能的遊戲解。

程式實現詳情

你必須編寫程式以實現以下的函數 (或方法):

- `string solve_puzzle(string s, int[] c)`.
 - s : 一個含有 n 個字符的字串。而對於每個 i ($0 \leq i \leq n - 1$) 字符 i 將會是以下其中一個:
 - 'X', 表示格位 i 必須要填上黑色,
 - '_', 表示格位 i 必須要填上白色,
 - '.', 表示格位 i 暫時沒有任何相關資料.
 - c : 是一個長度為 k 的陣列, 它的內容是上面敘述中所講的提示,
 - 這函數必須要返回一個長度為 n 的陣列。對於每個 i ($0 \leq i \leq n - 1$) 而言, 輸出字串的第 i 個字符應該是以下其中一個:
 - 'X', 表示在所有遊戲解中, 第 i 個格位都是填上黑色的,
 - '_', 表示在所有遊戲解中, 第 i 個格位都是填上白色的,
 - '?', 表示其他情況 (即是該次遊戲中存在兩個遊戲解, 且其中一個該格位是填上黑色, 但在另一遊戲解中該格位則是填上白色)。

在 C 語言中, 函數的標頭是略有不同, 其分別如下:

- `void solve_puzzle(int n, char* s, int k, int* c, char* result)`
 - `n`: 字串 `s` 的長度 (即格位的數目),
 - `k`: 陣列 `c` 的長度 (即提示的數目),
 - 其他的參數與上面的相同,
 - 這函數應把答案寫到字串 `result` 內, 而不是返回一個長度為 `n` 的字串。

在本題目中使用的字符其 ASCII 代碼如下:

- `'X'`: 88,
- `'_'`: 95,
- `'.'`: 46,
- `'?'`: 63.

請使用系統提供的程式模版以取得更多關於你所使用的編程語言的資料。

樣例

樣例 1

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

以下是這次遊戲的所有正確的填色方法:

- `"XXX_XXXX_"`,
- `"XXX__XXXX_"`,
- `"XXX___XXXX"`,
- `"_XXX_XXXX_"`,
- `"_XXX__XXXX"`,
- `"__XXX_XXXX"`.

你可以從以上觀察到序號為 2, 6, 及 7 (由 0 開始) 的格位在所有正確的填色方法中都是黑色。而其他的所有格位可能但不一定是黑色的。所以正確的答案應該是 `"??X???XX??"`。

樣例 2

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

在這個樣例中, 正確的填色方法只有一個, 所以正確的答案應該是 `"XXX_XXXX"`。

樣例 3

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

在這個樣例中, 我們可以知道格位 4 應該都是白色的, 因為在格位 3 和 5 之間根本沒可能容納 3 個連續的黑色。因此正確的答案應該是 `"???__????"`。

樣例 4

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

在這個樣例中共有以下兩個正確的填色方法是可以符合以上表述的:

- `"XXX_____"`,
- `"_XXX_____"`.

因此正確的答案是 `"?XX?_____"`。

子任務

在所有的子任務中 $1 \leq k \leq n$, 且對於每個 $0 \leq i \leq k-1$ 而言, $1 \leq c_i \leq n$.

1. (7 分) $n \leq 20$, $k = 1$, s 只含有字符 '.' (即是空白的遊戲行格),
2. (3 分) $n \leq 20$, s 只含有字符 '.',
3. (22 分) $n \leq 100$, s 只含有字符 '.',
4. (27 分) $n \leq 100$, s 只含有字符 '.' 及 '_' (只有關於填上了白色的格位資料),
5. (21 分) $n \leq 100$,
6. (10 分) $n \leq 5\,000$, $k \leq 100$,
7. (10 分) $n \leq 200\,000$, $k \leq 100$.

樣例評測程式

樣例評測程式將會讀取以下格式的輸入資料:

- 第一行: 字符串 s ,
- 第二行: 整數 k 隨後有 k 個整數 c_0, \dots, c_{k-1} .