

Paint By Numbers

Paint by Numbers es un juego de rompecabezas bien conocido. Consideremos una versión simple uni-dimensional de este rompecabezas. En este rompecabezas, al jugador le es dada una fila de n celdas. Las celdas están numeradas desde 0 hasta $n - 1$ de izquierda a derecha. El jugador debe pintar cada celda de blanco o de negro. Usaremos 'X' para denotar celdas negras y '_' para denotar celdas blancas.

Al jugador le es dada una secuencia $c = [c_0, \dots, c_{k-1}]$ de k enteros positivos: las *pistas*. Él debe pintar las celdas de tal manera que las celdas negras en la fila formen exactamente k bloques de celdas consecutivas. Además, el número de celdas negras en el i -ésimo bloque (base 0) de izquierda a derecha debe ser igual a c_i . Por ejemplo, si las pistas son $c = [3, 4]$, el rompecabezas resuelto debe tener exactamente dos bloques de celdas negras consecutivas: uno de longitud 3 y otro de longitud 4. Por ende, si $n = 10$ y $c = [3, 4]$, una solución que satisface las pistas es "XXX XXXX". Nota que "XXXX XXX" no satisface las pistas porque los bloques de celdas negras no están en el orden correcto. Asimismo, "XXXXXXXX" no satisface las pistas porque hay un solo bloque de celdas negras, no dos bloques separados.

Se te da un rompecabezas Paint by Numbers parcialmente resuelto. Esto significa que conoces n y c , y adicionalmente sabes que algunas celdas deben ser negras y otras celdas deben ser blancas. Tu tarea es deducir información adicional sobre las celdas.

Específicamente, una *solución válida* que satisfaga las pistas, y que también coincida con el color de las celdas conocidas. Tu programa debe encontrar celdas que son pintadas de negro en cada solución válida, y celdas que son pintadas de blanco en cada solución válida.

Puedes asumir que la entrada es tal que existe por lo menos una solución válida.

Detalles de Implementación

Debes implementar la siguiente función (método):

- `string solve_puzzle(string s, int[] c)`.
 - s : cadena de longitud n . Para cada i ($0 \leq i \leq n - 1$), el caracter i es:
 - 'X', si la celda i debe ser negra,
 - '_', si la celda i debe ser blanca,
 - '.', si no hay información acerca de la celda i .
 - c : arreglo de longitud k que contiene las pistas de acuerdo a lo definido arriba,
 - la función debe retornar una cadena de longitud n . Para cada i (

$0 \leq i \leq n - 1$), el caracter i de la cadena de salida debe ser :

- 'X', si la celda i es negra en cada solución válida,
- '_', si la celda i es blanca en cada solución válida,
- '?', de lo contrario (por ejemplo, si existen dos soluciones válidas de manera que la celda i es negra en una de ellas y blanca en otra).

En lenguaje C la firma de la función es ligeramente diferente:

- `void solve_puzzle(int n, char* s, int k, int* c, char* result)`
 - n : longitud de la cadena s (número de celdas),
 - k : longitud del arreglo c (número de pistas),
 - los demás parámetros son iguales a los de arriba,
 - en lugar de retornar una cadena de n caracteres, a función debe escribir la respuesta a la cadena `result`.

Ejemplos

Ejemplo 1

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

Estas son las todas las posibles soluciones válidas para este rombecabezas:

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

Puedes observar que las celdas con (base-0) índices 2,6, y 7 son negras en cada solución válida. Cada una de las otras celdas puede ser, pero no tiene que ser negra. Por ende, la respuesta correcta es "??X???XX??".

Ejemplo 2

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

En este ejemplo, la solución entera se determina de forma única y la respuesta correcta es "XXX_XXXX".

Ejemplo 3

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

En este ejemplo podemos deducir que la celda 4 debe ser blanca también - no hay forma de encajar tres celdas negras consecutivas entre las celdas blancas con índices 3 y 5.

Por ende, la respuesta correcta es "???__????".

Ejemplo 4

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

Sólo hay dos soluciones válidas que responden a la descripción anterior:

- "XXX_____",
- "_XXX_____".

Por ende, la respuesta correcta es "?XX?_____".

Subtareas

En todas las subtareas $1 \leq k \leq n$, y $1 \leq c_i \leq n$ para cada $0 \leq i \leq k - 1$.

1. (7 puntos) $n \leq 20$, $k = 1$, s contiene solo '.' (rompecabezas vacío),
2. (3 puntos) $n \leq 20$, s contiene solo '.',
3. (22 puntos) $n \leq 100$, s contiene solo '.',
4. (27 puntos) $n \leq 100$, s contiene solo '.' y '_' (información solo sobre celdas blancas),
5. (21 puntos) $n \leq 100$,
6. (10 puntos) $n \leq 5\,000$, $k \leq 100$,
7. (10 puntos) $n \leq 200\,000$, $k \leq 100$.

Grader de ejemplo

El grader de ejemplo lee la entrada en el siguiente formato:

- línea 1: cadena s ,
- línea 2: entero k seguido de k enteros c_0, \dots, c_{k-1} .