



Verven met getallen

'Verven met getallen' is een bekend puzzel spelletje. We beschouwen een simpele één-dimensionale versie van deze puzzel. In deze puzzel wordt aan de speler een rij van n cellen gegeven. De cellen zijn genummerd van 0 tot en met $n - 1$ van links naar rechts. De speler moet iedere cel zwart of wit verven. We gebruiken 'X' om zwarte cellen aan te geven en '_' om witte cellen aan te geven.

De speler krijgt een reeks $c = [c_0, \dots, c_{k-1}]$ van k positieve integers: de *hints*. Hij moet de cellen op zo een manier verven dat de zwarte cellen in de rij precies k blokken vormen van opeenvolgende cellen. Bovendien moet het aantal zwarte cellen in het i 'de blok van links (0-based) gelijk zijn aan c_i . Bijvoorbeeld, als de hints $c = [3, 4]$ zijn, moet de opgeloste puzzel precies twee blokken van opeenvolgende zwarte cellen hebben: een van lengte 3 gevolgd door een ander van lengte 4. Dus, als $n = 10$ en $c = [3, 4]$, is "`___XXX__XXXX`" een oplossing die aan de hints voldoet. Merk op dat "`XXXX__XXX__`" niet aan de hints voldoet: de blokken van zwarte cellen staan niet in de juiste volgorde. Tevens voldoet "`__XXXXXXXX__`" niet aan de hints: er is een enkel blok van zwarte cellen, niet twee aparte blokken.

Je krijgt een gedeeltelijk opgeloste 'Verven met getallen' puzzel. Dit betekent dat je n en c weet, en bovendien ook weet dat sommige cellen zwart moeten zijn en sommige cellen wit moeten zijn. Jouw taak is om extra informatie af te leiden over de cellen.

Meer specifiek, een geldige oplossing is een die en aan de hints voldoet en aan de gegeven cellen. Je programma moet cellen vinden die zwart geverfd zijn in elke geldige oplossing en cellen die wit geverfd zijn in elke geldige oplossing. Je mag aannemen dat de invoer zodanig is dat er minstens één geldige oplossing bestaat.

Implementatie details

Je moet de volgende functie (methode) implementeren:

- `string solve_puzzle(string s, int[] c)`.
 - s : string van lengte n . Voor elke i ($0 \leq i \leq n - 1$) is het i 'de character:
 - 'X', als cel i zwart moet zijn,
 - '_', als cel i wit moet zijn,
 - '.', als er geen informatie is over cel i .
 - c : array van lengte k die de hints bevat, zoals hierboven gedefinieerd,
 - de functie moet een string van lengte n terug geven. Voor elke i ($0 \leq i \leq n - 1$) moet het i 'de character van de uitvoer string gelijk zijn aan:
 - 'X', als cel i zwart is in elke geldige oplossing,
 - '_', als cel i wit is in elke geldige oplossing,

- '?', anders (dit betekent: als er twee geldige oplossingen bestaan zo dat cel i zwart is in de ene oplossing en wit in de andere oplossing).

In de taal C is de functie signature een beetje anders:

- `void solve_puzzle(int n, char* s, int k, int* c, char* result)`
 - `n`: lengte van de string `s` (aantal cellen),
 - `k`: lengte van de array `c` (aantal hints),
 - de andere parameters zijn hetzelfde als hierboven,
 - In de plaats van een string van n characters terug te geven, moet de functie het antwoord schrijven naar de string `result`.

De ASCII codes van de characters gebruikt in dit probleem zijn:

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

Gelieve de voorziene template bestanden te gebruiken voor implementatiedetails in jouw programmeertaal.

Voorbeelden

Voorbeeld 1

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

Dit zijn alle mogelijke geldige oplossingen van de puzzel:

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

Men kan opmerken dat de cellen met (0-based) indices 2, 6, en 7 zwart zijn in elke geldige oplossing. Elke andere cel kan, maar moet niet, zwart zijn. Dus, het juiste antwoord is "??X???XX??".

Voorbeeld 2

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

In dit voorbeeld is de hele oplossing uniek bepaald en het juiste antwoord is "XXX_XXXX".

Voorbeeld 3

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

In dit voorbeeld kunnen we afleiden dat cel 4 ook wit moet zijn — er is geen manier om drie opeenvolgende zwarte cellen te plaatsen tussen de witte cellen op indices 3 en 5. Dus, het juiste antwoord is "???__????".

Voorbeeld 4

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

Er zijn slechts twee geldige oplossingen die overeenkomen met bovenstaande beschrijving:

- "XXX_____",
- "_XXX_____".

Dus, het juiste antwoord is "?XX?_____".

Subtaken

In alle subtaken geldt $1 \leq k \leq n$, en $1 \leq c_i \leq n$ voor alle $0 \leq i \leq k - 1$.

1. (7 punten) $n \leq 20$, $k = 1$, s bevat enkel '.' (lege puzzel),
2. (3 punten) $n \leq 20$, s bevat enkel '.',
3. (22 punten) $n \leq 100$, s bevat enkel '.',
4. (27 punten) $n \leq 100$, s bevat enkel '.' en '_' (alleen informatie over witte cellen),
5. (21 punten) $n \leq 100$,
6. (10 punten) $n \leq 5\,000$, $k \leq 100$,
7. (10 punten) $n \leq 200\,000$, $k \leq 100$.

Voorbeeldgrader

De voorbeeldgrader leest de invoer in het volgende formaat:

- regel 1: string s ,
- regel 2: integer k gevolgd door k integers c_0, \dots, c_{k-1} .