Obrazek logiczny (Paint By Numbers)

Obrazki logiczne to znany typ łamigłówki. W tym zadaniu rozważamy jej prostą, jednowymiarową wersję. Rozwiązujący ma przed sobą rząd n pól. Pola te są ponumerowane od 0 do $n-1$ od lewej do prawej. Zadaniem rozwiązującego jest pokolorować pola na biało i czarno. Czarne pola oznaczamy jako 'X', a białe jako '_'.

Rozwiązujący ma zadany ciąg $c = [c_0, \ldots, c_{k-1}]$ złożony z k dodatnich liczb całkowitych, który nazywamy *wskazówką*. Jego zadaniem jest pokolorować pola w taki sposób, aby czarne pola tworzyły dokładnie k bloków kolejnych pól. Ponadto liczba czarnych pól w i-tym bloku od lewej (bloki numerujemy od 0) musi być równa c_i. Przykładowo, jeśli wskazówka to $c = [3, 4]$, to rozwiązanie łamigłówki musi zawierać dwa bloki czarnych pól: jeden o długości 3 i dalej drugi o długości 4. Tak więc jeśli $n = 10$ i $c = [3, 4]$, jednym z rozwiązań spełniających wymagania wskazówki jest "_XXX__XXXX". Zauważmy, że "XXXX_XXX__" nie spełnia wymagań wskazówki, jako że bloki czarnych pól znajdują się w złej kolejności. Także "__XXXXXXX_" nie spełnia wymagań wskazówki, gdyż zawiera tylko jeden blok czarnych pól, a nie dwa osobne.

Masz dany częściowo rozwiązany obrazek logiczny, tzn. znasz n i c i wiesz, że niektóre pola muszą być czarne, a niektóre białe. Twoim zadaniem jest wydedukować coś więcej na temat pól.

Mianowicie, przez *poprawne rozwiązanie* rozumiemy rozwiązanie spełniające wymagania wskazówki, które ponadto spełnia wymagania dotyczące znanych kolorów pól. Twój program powinien stwierdzić, które pola w dowolnym poprawnym rozwiązaniu będą pokolorowane na czarno i które pola w dowolnym poprawnym rozwiązaniu będą białe.

Możesz założyć, że wejście jest dobrane w taki sposób, że istnieje co najmniej jedno poprawne rozwiązanie.

Szczegóły implementacji

Powinieneś napisać jedną funkcję (metodę):

- `string solve_puzzle(string s, int[] c)`.
  - `s`: napis o długości n. Dla każdego i ($0 \leq i \leq n-1$), znak i to:
    - 'X', jeśli pole i musi być czarne,
    - '_', jeśli pole i musi być białe,
    - '.', jeśli nic nie wiadomo o polu i.
  - `c`: tablica rozmiaru k zawierająca wskazówkę, zdefiniowana powyżej.
  - Funkcja powinna zwrócić napis długości n. Dla każdego i ($0 \leq i \leq n-1$), znak i wynikowego napisu powinien być równy:

    - ‘X’, jeśli pole i jest czarne w każdym poprawnym rozwiązaniu,
    - ‘_’, jeśli pole i jest białe w każdym poprawnym rozwiązaniu,
    - ‘?’, w przeciwnym przypadku (tzn. jeśli istnieją dwa poprawne rozwiązania, takie że w pierwszym z nich pole i jest czarne, a w drugim białe).

W języku C sygnatura funkcji jest minimalnie inna:

- `void solve_puzzle(int n, char* s, int k, int* c, char* result)`
    - `n`: długość napisu `s` (liczba pól),
    - `k`: rozmiar tablicy `c` (długość wskazówki),
    - pozostałe parametry są takie same jak powyżej,
    - zamiast zwracać napis złożony z n znaków, funkcja powinna go zapisać do napisu `result`.

Kody ASCII znaków występujących w tym zadaniu to:

- ‘X’: 88,
- ‘_’: 95,
- ‘.’: 46,
- ‘?’: 63.

Szczegóły implementacji w Twoim języku programowania znajdują się w dostarczonych plikach z szablonami.

Przykłady

Przykład 1

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

Oto wszystkie poprawne rozwiązania łamigłówki:

- “XXX_XXXX__”,
- “XXX__XXXX_”,
- “XXX___XXXX”,
- “_XXX_XXXX_”,
- “_XXX__XXXX”,
- “__XXX_XXXX”.

Można zauważyć, że pola o indeksach (numerowanych od 0) 2, 6 i 7 w każdym poprawnym rozwiązaniu są czarne. Każde inne pole może, ale nie musi być czarne. Poprawną odpowiedzią jest zatem “`??X???XX??`”.

Przykład 2

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

W tym przykładzie całe rozwiązanie jest wyznaczone jednoznacznie i poprawną odpowiedzią jest “`XXX_XXXX`”.

Przykład 3

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

W tym przykładzie możemy wywnioskować, że pole o indeksie 4 musi być białe - nie ma możliwości pokolorowania trzech kolejnych pól na czarno pomiędzy białymi polami o indeksach 3 i 5. Zatem poprawną odpowiedzią jest "`???___????`".

Przykład 4

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

Mamy jednie dwa poprawne rozwiązania spełniające powyższy opis:

- "`XXX_______`",
- "`_XXX______`".

Tak więc poprawną odpowiedzią jest "`?XX?______`".

Podzadania

We wszystkich podzadaniach $1 \leq k \leq n$ oraz $1 \leq c_i \leq n$ dla każdego $0 \leq i \leq k-1$.

1. (7 punktów) $n \leq 20$, $k = 1$, s zawiera jedynie '.' (pusta łamigłówka),
2. (3 punkty) $n \leq 20$, s zawiera jedynie '.',
3. (22 punkty) $n \leq 100$, s zawiera jedynie '.',
4. (27 punktów) $n \leq 100$, s zawiera jedynie '.' oraz '_' (są to jedynie informacje o białych polach),
5. (21 punktów) $n \leq 100$,
6. (10 punktów) $n \leq 5\,000$, $k \leq 100$,
7. (10 punktów) $n \leq 200\,000$, $k \leq 100$.

Przykładowy program sprawdzający

Przykładowy program sprawdzający wczytuje dane w następującym formacie:

- wiersz 1: napis s,
- wiersz 2: liczba k, po której następuje k liczb całkowitych $c_0, \ldots, c_{k-1}$.