

## Emove krížovky

Určite poznáte tú situáciu, keď pri riešení hlavolamov hľadáte čoraz ťažšie a ťažšie úlohy, pretože sa pri riešení zlepšujete. To je tým, že ste šikovní. S Emom to ale také ľahké nie je. Preto väčšinou hľadá ľahšie a ľahšie hlavolamy, až kým nenájde také, ktoré už konečne zvládne vyriešiť. Najnovšie našiel jednorozmerné maľované krížovky.

1D maľovaná krížovka sa skladá z jedného riadka s  $n$  políčkami, ktoré sú zľava doprava očíslované od  $0$  po  $n - 1$ . Úlohou riešiteľa je zafarbiť niektoré políčka na čierne (označujeme znakom 'X') a niektoré nechať biele (označujeme znakom '\_'). Ofarbenie navyše musí spĺňať zadané podmienky. Podmienky sú udané v tvare  $c = [c_0, \dots, c_{k-1}]$ , kde  $k$  je kladné celé číslo. 1D krížovka je správne vyriešená vtedy, ak čierne políčka tvoria práve  $k$  súvislých úsekov, pričom  $i$ -ty čierny úsek zľava (čísľujúc od 0) obsahuje presne  $c_i$  políčok.

Napríklad, ak majú podmienky tvar  $c = [3, 4]$ , tak má maľovaná krížovka obsahovať obsahovať dva čierne úseky: prvý dĺžky 3 a druhý dĺžky 4. Ak by bolo  $n = 10$ , tak jedno možné riešenie vyzerá nasledovne: "XXX XXXX". Naopak, **nesprávne** je riešenie "XXXX XXX", pretože úsek dĺžky 4 je pred úsekom dĺžky 3. Takisto **nesprávne** je riešenie "XXXXXXXX", keďže sa skladá iba z jediného čierneho úseku.

Emova 1D maľovaná krížovka je už sčasti vyplnená. Vy dostanete popis toho ako je vyplnená, teda informácie o tom, ktoré políčka musia byť čierne a ktoré políčka musia byť biele. Taktiež dostanete  $c$ , čiže zoznam podmienok. Vašou úlohou je pomôcť Emovi a doplniť všetky ďalšie informácie, ktoré zo zadania logicky vyplývajú. Presnejšie, vašou úlohou je zafarbiť na čierne všetky políčka, ktoré sú určite čierne (t.j. majú čiernu farbu v každom možnom riešení) a na bielo všetky políčka, ktoré sú určite biele. Prekvapivo, Emo sa ešte nepomýlil, preto môžete predpokladať, že sa každá zadaná maľovaná krížovka dá korektne doplniť aspoň jedným spôsobom.

### Implementačné detaily

Vašou úlohou je naprogramovať funkciu:

- `string solve_puzzle(string s, int[] c)`.
  - $s$ : reťazec dĺžky  $n$ . Pritom  $i$ -ty znak tohto reťazca, pre  $0 \leq i \leq n - 1$ , je:
    - 'X', ak musí byť  $i$ -te políčko čierne,
    - '\_', ak musí byť  $i$ -te políčko biele,
    - '.', ak o  $i$ -tom políčku nemáme žiadnu dodatočnú informáciu.
  - $c$ : pole dĺžky  $k$  obsahujúce požadované dĺžky čiernych úsekov.
  - Funkcia by mala vrátiť reťazec dĺžky  $n$ . Pritom  $i$ -ty znak tohto reťazca, pre  $0 \leq i \leq n - 1$ , by mal byť:
    - 'X', ak je  $i$ -te políčko čierne v každom prípustnom riešení,

- `'_'`, ak je  $i$ -te políčko biele v každom prípustnom riešení,
- `'?'`, v opačnom prípade (teda ak existujú dve rôzne správne riešenia, v ktorých je  $i$ -te políčko zafarbené rôzne).

Pre jazyk C sa deklarácia funkcie trochu líši:

- `void solve_puzzle(int n, char* s, int k, int* c, char* result)`
  - $n$ : dĺžka reťazca  $s$  (počet políčok maľovanej krížovky),
  - $k$ : dĺžka pola  $c$  (počet podmienok),
  - ostatné parametre sú rovnaké ako predtým,
  - namiesto vrátenia reťazca dĺžky  $n$  by funkcia mala riešenie zapísať do reťazca `result`.

Znaky použité v úlohe majú nasledovné ASCII hodnoty:

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

Pre lepšie pochopenie konkrétnej implementácie vo vami zvolenom jazyku nahliadnite do priložených ukážkových súborov.

## Príklady

### Príklad 1

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

Nižšie sú vypísané všetky správne riešenia tejto maľovanej krížovky:

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

Môžete si všimnúť, že políčka na pozíciách 2, 6 a 7 (čísľujúc od 0) sú čierne vo všetkých správnych riešeniach. Všetky ostatné políčka môžu, ale nemusia, byť čierne, preto o nich nevieme povedať nič. Správne riešenie je preto reťazec `"??X???XX??"`.

### Príklad 2

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

V tomto príklade existuje jediné správne riešenie, ktoré vyzerá `"XXX_XXXX"`. Takýto reťazec by mala vrátiť aj volaná funkcia. Všimnite si, že vieme určiť úplne celé riešenie, vrátane bieleho políčka.

### Príklad 3

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

V tomto príklade si môžeme logicky odvodiť, že políčko 4 musí byť biele, keďže

nemôžeme mať súvislý čierny úsek dĺžky 3 medzi zaručene bielymi políčkami 3 a 5. Odpoveď je teda “**???**\_\_**????**”.

#### Príklad 4

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

Existujú iba dve správne riešenia:

- “**XXX**\_\_\_\_\_”,
- “\_**XXX**\_\_\_\_\_”.

Preto je správna odpoveď “**?XX?**\_\_\_\_\_”.

#### Podproblémy

Vo všetkých podúlohách môžete predpokladať, že  $1 \leq k \leq n$  a  $1 \leq c_i \leq n$  pre všetky  $0 \leq i \leq k - 1$ .

Pre jednotlivé podúlohy platia navyše nasledujúce obmedzenia:

1. (7 bodov)  $n \leq 20$ ,  $k = 1$ ,  $s$  obsahuje iba znaky ‘.’ (ešte nemáme nič predvyplnené),
2. (3 body)  $n \leq 20$ ,  $s$  obsahuje iba znaky ‘.’,
3. (22 bodov)  $n \leq 100$ ,  $s$  obsahuje iba znaky ‘.’,
4. (27 bodov)  $n \leq 100$ ,  $s$  obsahuje iba znaky ‘.’ a ‘\_’ (teda o niektorých políčkach vieme, že sú biele),
5. (21 bodov)  $n \leq 100$ ,
6. (10 bodov)  $n \leq 5\,000$ ,  $k \leq 100$ ,
7. (10 bodov)  $n \leq 200\,000$ ,  $k \leq 100$ .

#### Ukázkový grader

Ukázkový grader, ktorý máte k dispozícii číta vstup v nasledovnom formáte:

- riadok 1: reťazec  $s$ ,
- riadok 2: číslo  $k$  nasledované  $k$  číslami  $c_0, \dots, c_{k-1}$ .