

## Slikanje brojevima

"Slikanje brojevima" je u nekim zemljama veoma popularna igra. U ovom zadatku razmotrit ćemo jednostavnu jednodimenzionalnu verziju ove igre. U našoj verziji, igraču je na raspolaganju jedan red koji sadrži  $n$  polja. Polja su označena s lijeva na desno redom od 0 do  $n - 1$ . Igrač mora obojiti svako polje u redu crnom ili bijelom bojom. Crna polja označavamo sa 'X' dok bijela polja označavamo sa '\_'.

Igraču su dani *tragovi* predstavljeni nizom  $k$  pozitivnih cijelih brojeva  $c = [c_0, \dots, c_{k-1}]$ . Igrač mora obojiti polja na način da crna polja u redu formiraju tačno  $k$  blokova uzastopnih polja. Štaviše, broj crnih polja u  $i$ -tom bloku s lijeva (brojeći od 0) mora biti jednak  $c_i$ . Na primjer, ako su tragovi  $c = [3, 4]$ , rješenje igre mora imati tačno dva bloka uzastopnih crnih polja: prvi dužine 3 i drugi dužine 4. Odatle, ako je  $n = 10$  i  $c = [3, 4]$ , jedno valjano rješenje igre je "XXXXXXX". Uočite da "XXXXXXX " nije valjano rješenje: blokovi uzastopnih crnih polja nisu u pravom poretku. Također, "XXXXXXXX" nije valjano rješenje jer postoji samo jedan blok uzastopnih crnih polja, a ne dva odvojena bloka.

Zadana je djelomično riješena igra "slikanje brojevima". Poznati su vam  $n$  i  $c$  i dodatno znate da određena polja moraju biti crne boje i da određena polja moraju biti bijele boje. Vaš je zadatak izvesti dodatne zaključke o poljima.

Preciznije, ako *valjanim* nazivamo rješenje koje zadovoljava tragove te odgovara bojama poznatih polja, morate odrediti koja će polja biti crne boje u svakom valjanom rješenju igre i koja će polja biti bijele boje u svakom valjanom rješenju igre. Možete pretpostaviti da će ulazni podaci biti takvi da uvijek postoji bar jedno valjano rješenje igre.

### Implementacijski detalji

Potrebno je implementirati sljedeću funkciju (metodu):

- `string solve_puzzle(string s, int[] c)`.
  - $s$ : string dužine  $n$ . Za svaki  $i$  ( $0 \leq i \leq n - 1$ ) znak  $i$  je:
    - 'X', ako polje  $i$  mora biti crne boje,
    - '\_', ako polje  $i$  mora biti bijele boje,
    - '.', ako nema informacije o polju  $i$ .
  - $c$ : niz dužine  $k$  koji sadrži tragove, kako je definirano ranije.
  - funkcija vraća string dužine  $n$ . Za svaki  $i$  ( $0 \leq i \leq n - 1$ ) znak  $i$  izlaznog stringa mora biti:
    - 'X', ako je polje  $i$  crne boje u svakom valjanom rješenju igre,
    - '\_', ako je polje  $i$  bijele boje u svakom valjanom rješenju igre,
    - '?', u ostalim slučajevima (tj., ako postoje dva valjana rješenja igre

takva da je polje  $i$  u jednom od tih rješenja crne boje a u drugom rješenju bijele boje).

U programskom jeziku C prototip je:

- `void solve_puzzle(int n, char* s, int k, int* c, char* result)`
  - `n`: dužina stringa `s` (broj polja),
  - `k`: dužina niza `c` (broj tragova),
  - ostali parametri isti su kao gore,
  - umjesto da vrati string od `n` znakova, funkcija upisuje odgovor u string `result`.

ASCII kodovi znakova korištenih u ovom zadatku:

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

Za implementacijske detalje koristite zadate template datoteke.

## Primjeri

### Primjer 1

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

Sva su moguća rješenja igre sljedeća:

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

Uočite da su, u svim rješenjima igre, polja s (nula-indeksiranim) indeksima 2, 6 i 7 crne boje. Sva ostala polja mogu ali i ne moraju biti crne boje. Dakle, tačan je odgovor `"??X???XX??"`.

### Primjer 2

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

U ovom primjeru postoji tačno jedno rješenje `"XXX_XXXX"`.

### Primjer 3

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

U ovom primjeru možete zaključiti da polje 4 također mora biti bijele boje — ne postoji način da postavite uzastopna crna polja između bijelih polja na pozicijama (indeksima) 3 i 5. Stoga je tačan odgovor `"???__????"`.

### Primjer 4

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

Postoje samo dva valjana rješenja igre koja odgovaraju danom opisu:

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

Stoga, tačan je odgovor `"?XX?_____"`.

## Podzadaci

U svim je podzadacima  $1 \leq k \leq n$  te  $1 \leq c_i \leq n$  za svaki  $0 \leq i \leq k - 1$ .

1. (7 bodova)  $n \leq 20$ ,  $k = 1$ ,  $s$  sadrži samo `'.'` (prazna igra),
2. (3 boda)  $n \leq 20$ ,  $s$  sadrži samo `'.'`,
3. (22 boda)  $n \leq 100$ ,  $s$  sadrži samo `'.'`,
4. (27 bodova)  $n \leq 100$ ,  $s$  sadrži samo `'.'` i `'_'` (informacije samo o bijelim poljima),
5. (21 bod)  $n \leq 100$ ,
6. (10 bodova)  $n \leq 5\,000$ ,  $k \leq 100$ ,
7. (10 bodova)  $n \leq 200\,000$ ,  $k \leq 100$ .

## Priloženi grader

Priloženi grader učitava podatke sa standardnog ulaza u sljedećem obliku:

- redak 1: string  $s$ ,
- redak 2: cijeli broj  $k$  za kojim slijedi  $k$  cijelih brojeva  $c_0, \dots, c_{k-1}$ .