

## Jaapani mõistatus

Jaapani mõistatus on üsna tuntud nuputusmäng. Me vaatleme selle mängu ühemõõtmelist versiooni. Selles versioonis antakse mängijale  $n$  ruudust koosnev rida. Ruudud on nummerdatud vasakult paremale  $0 \dots n - 1$ . Mängija peab värvima iga ruudu kas mustaks või valgeks. Me tähistame musti ruute 'X'-ga ja valgeid '\_'-ga.

Mängijale antakse ka  $k$  positiivsest täisarvust koosnev jada  $c = [c_0, \dots, c_{k-1}]$ , mida nimetatakse *vihjeteks*. Mängija peab ruudud värvima nii, et mustad ruudud moodustavat täpselt  $k$  järjestikuste ruutude plokki. Lisaks peab vasakult lugedes  $i$ . plokk (numeratsioon algab nullist) sisaldama täpselt  $c_i$  ruutu. Näiteks, kui vihjed on  $c = [3, 4]$ , peab mõistatuse lahenduses olema täpselt kaks mustade ruutude plokki, mis koosnevad vastavalt 3-st ja 4-st ruudust. Seega, kui  $n = 10$  and  $c = [3, 4]$ , on üks vihjetega sobiv lahendus "XXX XXXX". Pane tähele, et "XXXX XXX" ei ole vihjetega sobiv lahendus, sest plokkid on vales järjekorras. "XXXXXXXX" ei ole ka vihjetega sobiv lahendus, sest siin on ainult üks pikk plokk, mitte kaks eraldi plokki.

Sulle antakse osaliselt lahendatud mõistatus. See tähendab, et antud on  $n$  ja  $c$  ning lisaks on mõnede ruutude kohta teada, et need peavad olema mustad ja mõnede kohta, et need peavad olema valged. Sinu ülesanne on tuletada selle põhjal lisainfot ruutude värvide kohta.

Täpsemalt nimetame *täislahenduseks* mõistatuse lahendust, mis sobib vihjetega ja milles lisaks klapiivad ka kõigi teadaolevate ruutude värvid. Sinu programm peab tegema kindlaks, millised ruudud on igas täislahenduses kindlasti mustad ja millised on kindlasti valged.

Võib eeldada, et sisend on selline, et leidub vähemalt üks täislahendus.

### Realisatsioon

Sinu lahendus peab realiseerima järgmise funktsiooni:

- `string solve_puzzle(string s, int[] c)`
  - $s$ : sõne pikkusega  $n$ , milles  $i$ . märk (kus  $0 \leq i \leq n - 1$ ) on:
    - 'X', kui  $i$ . ruut peab kindlasti olema must;
    - '\_', kui  $i$ . ruut peab kindlasti olema valge;
    - '.', kui  $i$ . ruudu kohta andmed puuduvad;
  - $c$ : massiiv pikkusega  $k$ , mis sisaldab eelpool kirjeldatud vihjeid;
  - funktsioon peab tagastama sõne pikkusega  $n$ , milles  $i$ . märk (kus  $0 \leq i \leq n - 1$ ) on:
    - 'X', kui  $i$ . ruut on igas täislahenduses must;
    - '\_', kui  $i$ . ruut on igas täislahenduses valge;

- '?', kui leiduvad sellised kaks täislahendust, et *i*. ruut on ühes must ja teises valge.

C keeles on funktsiooni liides natuke teistsugune:

- `void solve_puzzle(int n, char* s, int k, int* c, char* result)`
  - *n*: sõne *s* pikkus (ruutude arv);
  - *k*: massiivi *c* pikkus (vihjete arv);
  - teised sisendparameetrid on samasugused nagu eelmisel juhul;
  - *n* märgiga sõne tagastamise asemel peab funktsioon salvestama vastuse sõnasse `result`.

Selles ülesandes kasutatud märkide ASCII koodid:

- 'X': 88;
- '\_': 95;
- '.': 46;
- '?': 63.

Vaata ka näitekoodi failides olevat keelespetsiifilist lisainfot.

## Näited

### Näide 1

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

Mõistatuse võimalikud täislahendused on:

- "XXX\_XXXX\_",
- "XXX\_\_XXXX\_",
- "XXX\_\_\_XXXX",
- "\_XXX\_XXXX\_",
- "\_XXX\_\_XXXX",
- "\_\_XXX\_XXXX".

Võib panna tähele, et ruudud (nullist algavate) indeksitega 2, 6 ja 7 on kõigis täislahendustes mustad. Ülejäänud ruudud on mõnes täislahenduses mustad ja mõnes mitte. Seega on õige vastus "??X???XX??".

### Näide 2

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

Selles näites on mõistatuse täislahendus ühene ja õige vastus seega "XXX\_XXXX".

### Näide 3

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

Selles näites me võime järeldada, et 4. ruut peab ka valge olema, sest kolm järjestikust musta ruutu ei saa kahe valge ruudu vahele mahtuda. Seega on õige vastus "???.\_\_????".

### Näide 4

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

Selles näites on mõistatusel ainult kaks võimalikku täislahendust:

- "XXX\_\_\_\_\_",
- " \_XXX\_\_\_\_\_".

Seega on õige vastus "?XX?\_\_\_\_\_".

## Alamülesanded

Kõigis alamülesannetes on  $1 \leq k \leq n$  ja iga  $0 \leq i \leq k - 1$  korral  $1 \leq c_i \leq n$ .

1. (7 punkti)  $n \leq 20$ ;  $k = 1$ ;  $s$  sisaldab ainult '.' (mänguväli on tühi).
2. (3 punkti)  $n \leq 20$ ;  $s$  sisaldab ainult '.'.
3. (22 punkti)  $n \leq 100$ ;  $s$  sisaldab ainult '.'.
4. (27 punkti)  $n \leq 100$ ;  $s$  sisaldab ainult '.' ja '\_' (info ainult valgete ruutude kohta).
5. (21 punkti)  $n \leq 100$ .
6. (10 punkti)  $n \leq 5\,000$ ;  $k \leq 100$ .
7. (10 punkti)  $n \leq 200\,000$ ;  $k \leq 100$ .

## Näitekood

Näitekood loeb sisendi järgmisel kujul:

- Rida 1: sõne  $s$ .
- Rida 2: täisarv  $k$  ja selle järel  $k$  täisarvu  $c_0, \dots, c_{k-1}$ .