Gobelin

Gobelin je zelo znana logična uganka. Omejili se bomo na enodimenzionalno inačico te uganke. Pri tej uganki dobi igralec vrstico dolžine n. Celice so oštevilčene od 0 do $n-1$, od leve proti desni. Igralec mora obarvati vsako celico ali s črno ali z belo. Z znakom 'X' označujemo črne celice in z '_' označujemo bele celice.

Igralec dobi niz pozitivnih celih števil $c = [c_0, \ldots, c_{k-1}]$, ki predstavlja k *namigov*. Obarvati mora celice na tak način, da črne celice tvorijo natanko k blokov zaporednih celic. Število črnih celic i-tega bloka z leve (štejemo od 0) mora biti c_i. Na primer, če so namigi $c = [3, 4]$, bo imela rešena uganka natanko dva bloka zaporednih črnih celic: enega dolžine 3 in za tem še enega dolžine 4.

Zato, če je $n = 10$ in $c = [3, 4]$, je ena veljavna rešitev, ki zadošča namigom, "_XXX__XXXX". Opomnimo, da rešitev "XXXX_XXX__" ne ustreza namigom, saj bloki črnih celic niso v pravilnem vrstnem redu. Tudi "__XXXXXXX_" ne ustreza namigom, ker vsebuje en blok črnih celic in ne dveh ločenih blokov.

Podan je delno rešen gobelin. Poznan je n in c, dodatno pa vemo, da morajo biti nekatere celice črne in nekatere bele. Tvoja naloga je izpeljati dodatne informacije.

Natančneje, *veljavna rešitev* zadošča namigom in je skladna z barvami znanih celic. Tvoj program mora posikati celice, ki so v vsaki veljavni rešitvi obarvane črno, in celice, ki so v vsaki veljavni rešitvi obarvane belo.

Lahko predpostaviš, da je vhod tak, da obstaja vsaj ena veljavna rešitev.

Podrobnosti implementacije

Implementirati moraš naslednjo funkcijo (metodo):

- `string solve_puzzle(string s, int[] c)`.
  - `s`: niz dolžine n. Za vsak i ($0 \leq i \leq n-1$) je i-ti znak:
    - 'X', če mora biti celica i črna,
    - '_', če mora biti celica i bela,
    - '.', če ne moremo ničesar sklepati o celici i.
  - `c`: polje dolžine k vsebuje namige, kot je definirano zgoraj,
  - funkcija mora vrniti niz dolžine n. Za vsak i ($0 \leq i \leq n-1$) mora biti i-ti znak izhodnega niza:
    - 'X', če je celica i črna v vsaki veljavni rešitvi,
    - '_', če je celica i bela v vsaki veljavni rešitvi,
    - '?', sicer (t.j. obstajata dve veljavni rešitvi, tako da je celica i črna v eni rešitvi in bela v drugi).

V programskem jeziku C je podpis funkcije nekoliko drugačen:

- `void solve_puzzle(int n, char* s, int k, int* c, char* result)`
  - `n`: dolžina niza `s` (število celic),
  - `k`: dolžina polja `c` (število namigov),
  - ostali parametri so enaki kot zgoraj,
  - namesto vračanja niza n znakov, mora funkcija zapisati rezultat v niz `result`.

ASCII kode uporabljenih znakov pri tej nalogi so:

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

Uporabi predložne datoteke za več informacij o implementaciji v izbranem programskem jeziku.

Primeri

1. primer

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

Vse veljavne rešitve uganke so:

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

Lahko opazimo, da so celice na mestih (štejemo od 0) 2, 6 in 7 črne v vsaki veljavni rešitvi. Vsaka izmed ostalih celic je lahko, vendar ni nujno, črna. Zato je pravilni odgovor "`??X???XX??`".

2. primer

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

V tem primeru je celotna rešitev enolično določena in pravilni odgovor je "`XXX_XXXX`".

3. primer

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

V tem primeru lahko sklepamo, da mora biti tudi celica 4 bela - ni načina, da lahko vstavimo tri zaporedne črne celice med bele celice na mestih 3 in 5. Zato je pravilni odgovor "`???___????`".

4. primer

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

Obstajata samo dve veljavni rešitvi, ki ustrezata zgornjemu opisu:

- "XXX_______",
- "_XXX______".

Zatorej je pravilni odgovor "?XX?_____".

Podnaloge

Pri vseh podnalogah velja $1 \leq k \leq n$ in $1 \leq c_i \leq n$ za vsak $0 \leq i \leq k-1$.

1. (7 točk) $n \leq 20$, $k = 1$, s vsebuje samo '.' (prazna uganka),
2. (3 točke) $n \leq 20$, s vsebuje samo '.',
3. (22 točk) $n \leq 100$, s vsebuje samo '.',
4. (27 točk) $n \leq 100$, s vsebuje samo '.' in '_' (informacije samo o belih celicah),
5. (21 točk) $n \leq 100$,
6. (10 točk) $n \leq 5\,000$, $k \leq 100$,
7. (10 točk) $n \leq 200\,000$, $k \leq 100$.

Vzorčni ocenjevalnik

Vzorčni ocenjevalnik bere vhod sledeče oblike:

- 1. vrstica: niz s,
- 2. vrstica: celo število k, kateremu sledi k celih števil $c_0, \ldots, c_{k-1}$.