

Paint By Numbers

Paint By Numbers je u nekim zemljama veoma popularna igra. U ovom zadatku razmatraćemo jednostavnu jednodimenzionalnu verziju ove igre. U našoj verziji, igraču je na raspolaganju jedan red koji sadrži n polja. Polja su numerisana slijeva udesno 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 dati *tragovi* predstavljeni nizom k pozitivnih cijelih brojeva $c = [c_0, \dots, c_{k-1}]$. Igrač mora obojiti polja na taj način da crna polja u redu formiraju tačno k blokova uzastopnih polja. Štaviše, broj crnih polja u i -tom bloku slijeva (ne zaboravite da su polja numerisana 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 blok dužine 3 i drugi blok dužine 4. Otuda, ako je $n = 10$ i $c = [3, 4]$, jedno validno rješenje igre koje zadovoljava date tragove je "XXX XXXX". Uočite da "XXXX XXX" nije validno rješenje koje zadovoljava date tragove: blokovi uzastopnih crnih polja nisu u pravom poretku. Takođe, "XXXXXXX" nije validno rješenje koje zadovoljava date tragove jer postoji samo jedan blok uzastopnih crnih polja, a ne dva odvojena bloka.

Zadata je djelimično riješena igra Paint By Numbers. Poznati su vam n i c i, dodatno, znate da određena polja moraju biti crne boje i da neka polja moraju biti bijele boje. Vaš zadatak je da izvedete dodatne zaključke o poljima igre. Konkretno, *validno rješenje igre* je ono koje zadovoljava sve tragove i na poljima na kojima je boja poznata dolazi do poklapanja boja. Vaš program mora odrediti koja će polja biti crne boje u svakom validnom rješenju igre i koja će polja biti bijele u svakom validnom rješenju igre.

Možete pretpostaviti da će ulazni podaci biti takvi da uvijek postoji bar jedno validno rješenje igre.

Detalji implementacije

Potrebno je implementirati sljedeću funkciju (method):

- `string solve_puzzle(string s, int[] c)`.
 - s : string dužine n . Za svako i ($0 \leq i \leq n - 1$) karakter 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 definisano ranije,
 - funkcija vraća string dužine n . Za svako i ($0 \leq i \leq n - 1$), karakter i izlaznog stringa mora biti:
 - 'X', ako je polje i crne boje u svakom validnom rješenju igre,

- `'_'`, ako je polje i bijele boje u svakom validnom rješenju igre,
- `'?'`, u ostalim slučajevima (tj., ako postoje dva validna rješenja igre takva da je polje i u jednom od tih rješenja crne boje a u drugom rješenju je bijele boje).

U programskom jeziku C signatura 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 su isti kao gore,
 - umjesto da vrati string od n karaktera, funkcija upisuje odgovor u string `result`.

ASCII kodovi karaktera iz ovog zadatka su:

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

Za detalje implementacije koristite date template datoteke.

Primjeri

Primjer 1

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

Sva moguća rješenja igre su:

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

Uočite da su, u svim rješenjima igre, polja sa indeksima 2, 6 i 7 crne boje (indeksi kreću od 0). Sva ostala polja, mogu ali i ne moraju biti crne boje. Dakle, tačan odgovor je `"??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 takođe mora biti bijele boje— ne postoji način da postavite uzastopna crna polja između bijelih polja na pozicijama (indeksima) 3 i 5. Otuda je tačan odgovor `"???__????"`.

Primjer 4

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

Postoje samo dva validna rješenja igre koja odgovaraju datom opisu:

- "XXX_____",
- " _XXX_____".

Otuda, tačan odgovor je "?XX?_____".

Podzadaci

U svim podzadacima važi $1 \leq k \leq n$ i $1 \leq c_i \leq n$ za svako $0 \leq i \leq k - 1$.

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) $n \leq 200\,000$, $k \leq 100$.

Sample grader

Program za ocjenjivanje učitava podatke sa standardnog ulaza u sljedećem formatu:

- red 1: string s ,
- line 2: cio broj k za kojim slijedi k cijelih brojeva c_0, \dots, c_{k-1} .