Paint By Numbers

Paint By Numbers je vrlo popularna logička igra koju cepaju Boromir i Džoni. U ovom zadatku razmatramo jednostavnu jednodimenzionalnu verziju ove igre. U našoj verziji, igraču je na raspolaganju jedan red koji sadrži n polja. Polja su numerisana sleva udesno, redom, od 0 do $n-1$. Igrač mora obojiti svako polje crnom ili belom bojom. Crna polja označavamo sa 'X' dok bela polja označavamo sa '_'.

Igraču su dati *tragovi* u obliku niza $c = [c_0, \ldots, c_{k-1}]$ koji se sastoji od k pozitivnih celih brojeva. Igrač mora obojiti polja tako da crne ćelije formiraju tačno k blokova uzastopnih ćelija. Štaviše, broj crnih polja u i-tom bloku sleva (sve numeracije su od 0) mora biti jednak c_i. Na primer, ako su tragovi $c = [3, 4]$, rešenje igre mora imati tačno dva bloka uzastopnih crnih polja: prvi dužine 3 i drugi dužine 4. Dakle, ako je $n = 10$ i $c = [3, 4]$, jedno rešenje igre koje je u saglasnosti sa tragovima je "_XXX__XXXX". Uočite da "XXXX_XXX__" nije u saglasnosti sa tragovima jer blokovi uzastopnih crnih polja nisu u pravom poretku. Takođe, "__XXXXXXX_" nije u saglasnosti sa tragovima jer postoji samo jedan blok uzastopnih crnih polja, a ne dva odvojena bloka.

Vama je data delimično rešena igra Paint By Numbers. Poznati su vam n i c i, dodatno, znate da određena polja moraju biti crne boje i da određena polja moraju biti bele boje. Vaš zadatak je da izvedete dodatne zaključke o poljima.

Preciznije, *validno rešenje* je ono koje je u saglasnosti sa tragovima i koje je takođe u saglasnosti sa bojama poznatih ćelija. Vaš program mora pronaći polja koja su obojena crnom bojom u svakom validnom rešenju i mora pronaći polja koja su obojena belom bojom u svakom validnom rešenju.

Možete pretpostaviti da će ulazni podaci biti takvi da će uvek postojati bar jedno validno rešenje igre.

Detalji implementacije

Potrebno je da implementirate sledeću funkciju:

- `string solve_puzzle(string s, int[] c)`.
  - `s`: string dužine n. Za svako i ($0 \leq i \leq n-1$) karakter na poziciji i je:
    - 'X', ako polje i mora biti crno,
    - '_', ako polje i mora biti belo,
    - '.', ako ne postoji informacija o polju i.
  - `c`: niz dužine k koji predstavlja tragove, koji su definisani ranije,
  - funkcija treba da vrati string dužine n. Za svako i ($0 \leq i \leq n-1$) i-ti karakter ovog stringa treba da bude:

    - ‘X’, ako je polje i crno u svakom validnom rešenju,
    - ‘_’, ako je polje i belo u svakom validnom rešenju,
    - ‘?’, inače (tj., ako postoje dva validna rešenja tako da je polje i u jednom od njih crno a u drugom belo).

U jeziku C je potpis funkcije malo drugačiji:

- `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 i pre,
    - umesto da vrati string dužine n, funkcija treba da upiše rešenje u string `result`.

ASCII kodovi karaktera korišćenih u zadatku:

- ‘X’: 88,
- ‘_’: 95,
- ‘.’: 46,
- ‘?’: 63.

Koristite date templejt-fajlove za bolji uvid u detalje implementacije za vaš programski jezik.

Primeri

Primer 1

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

Ovo su sva validna rešenja za ovaj primer

- “XXX_XXXX__”,
- “XXX__XXXX_”,
- “XXX___XXXX”,
- “_XXX_XXXX_”,
- “_XXX__XXXX”,
- “__XXX_XXXX”.

Uočite da su, u svim validnim rešenjima igre, polja sa indeksima 2, 6 i 7 crne boje. Sva ostala polja mogu ali i ne moraju biti crne boje. Dakle, tačan odgovor je “??X???XX??”.

Primer 2

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

U ovom primeru postoji tačno jedno validno rešenje i tačan odgovor je “XXX_XXXX”.

Primer 3

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

U ovom primeru možete zaključiti da polje 4 takođe mora biti bele boje— ne postoji način da postavite tri uzastopna crna polja između belih polja na pozicijama 3 i 5.

Otuda je tačan odgovor “`???___????`”.

Primer 4

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

Postoje tačno dva validna rešenja za ovaj primer:

- “`XXX_______`”,
- “`_XXX______`”.

Dakle, tačan odgovor je “`?XX?______`”.

Podzadaci

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

1. (7 poena) $n \leq 20$, $k = 1$, s sadrži samo ‘`.`’ (prazna igra),
2. (3 poena) $n \leq 20$, s sadrži samo ‘`.`’,
3. (22 poena) $n \leq 100$, s sadrži samo ‘`.`’,
4. (27 poena) $n \leq 100$, s sadrži samo ‘`.`’ i ‘`_`’ (informacije samo o belim poljima),
5. (21 poen) $n \leq 100$,
6. (10 poena) $n \leq 5\,000$, $k \leq 100$,
7. (10 poena) $n \leq 200\,000$, $k \leq 100$.

Opis priloženog grejdera

Priloženi grejder čita ulaz u sledećem formatu:

- linija 1: string s,
- linija 2: ceo broj k, a zatim k celih brojeva $c_0, \ldots, c_{k-1}$.