

Ζωγραφίζοντας με Αριθμούς

Το "Ζωγραφίζοντας με Αριθμούς" είναι ένα πολύ γνωστό παιχνίδι παζλ. Θεωρούμε μία απλή μονοδιάστατη έκδοση αυτού του παζλ. Σε αυτό το παζλ, δίνεται στον παίκτη μία σειρά από n κελιά (cells). Τα κελιά (cells) αριθμούνται από 0 έως $n - 1$ ξεκινώντας από τα αριστερά προς τα δεξιά. Ο παίκτης πρέπει να χρωματίσει κάθε κελί με μαύρο ή λευκό χρώμα. Χρησιμοποιούμε το 'X' για να συμβολίσουμε τα μαύρα κελιά (cells) και '_' για να συμβολίσουμε τα λευκά κελιά (cells).

Στον παίκτη δίνεται μια ακολουθία $c = [c_0, \dots, c_{k-1}]$ από k θετικούς ακεραίους: τις *ενδείξεις* (clues). Πρέπει να χρωματίσει τα κελιά με τέτοιο τρόπο, έτσι ώστε τα μαύρα κελιά στη γραμμή να σχηματίζουν ακριβώς k τμήματα (blocks) από συνεχόμενα κελιά (cells). Επιπλέον, το πλήθος των μαύρων κελιών (cells) στο i -στο τμήμα (block) (ξεκινώντας από το 0) από αριστερά, πρέπει να είναι ίσο με το c_i . Για παράδειγμα, εάν οι ενδείξεις (clues) είναι $c = [3, 4]$, το λυμένο παζλ πρέπει να έχει ακριβώς δύο τμήματα (blocks) από συνεχόμενα μαύρα κελιά (cells): ένα μήκους 3 και ένα ακόμη μήκους 4. Έτσι, εάν $n = 10$ και $c = [3, 4]$, μία λύση που ικανοποιεί τις ενδείξεις (clues) είναι η "_XXX_XXXX". Σημειώστε ότι η "XXXX_XXX_" δεν ικανοποιεί τις ενδείξεις (clues) γιατί τα τμήματα (blocks) των μαύρων κελιών (cells) δεν είναι σε σωστή σειρά. Επίσης, η "_XXXXXXXX_" δεν ικανοποιεί τις ενδείξεις (clues) διότι υπάρχει ένα μοναδικό τμήμα (block) από μαύρα κελιά, και όχι δύο ξεχωριστά τμήματα.

Σας δίνεται μία μερική λύση του παζλ "Ζωγραφίζοντας με Αριθμούς". Συγκεκριμένα: γνωρίζετε τα n και c και, επιπρόσθετα γνωρίζετε ότι μερικά κελιά (cells) πρέπει να είναι μαύρα και μερικά κελιά (cells) πρέπει να είναι λευκά. Στόχος σας είναι να εξάγετε επιπρόσθετες πληροφορίες σχετικά με τα κελιά (cells).

Συγκεκριμένα, μία *έγκυρη λύση* είναι αυτή που ικανοποιεί τις ενδείξεις (clues), και επίσης συμφωνεί με τον χρωματισμό των ήδη γνωστών κελιών (cells). Το πρόγραμμά σας πρέπει να βρίσκει τα κελιά (cells), τα οποία έχουν χρωματιστεί με μαύρο χρώμα σε κάθε έγκυρη λύση και τα κελιά (cells), τα οποία έχουν χρωματιστεί με λευκό χρώμα σε κάθε έγκυρη λύση.

Υποθέστε ότι η είσοδος είναι τέτοια ώστε να υπάρχει τουλάχιστον μία έγκυρη λύση.

Λεπτομέρειες Υλοποίησης

Πρέπει να υλοποιήσετε την παρακάτω συνάρτηση (method):

- `string solve_puzzle(string s, int[] c)`.
 - s : συμβολοσειρά μήκους n . Για κάθε i ($0 \leq i \leq n - 1$) ο χαρακτήρας i είναι:
 - 'X', εάν το κελί (cell) i πρέπει να είναι μαύρο,

- `'_'`, εάν το κελί (cell) i πρέπει να είναι λευκό,
- `'.'`, εάν δεν υπάρχει πληροφορία για το κελί (cell) i .
- **c**: πίνακας μεγέθους k που περιέχει τις ενδείξεις (clues), όπως ορίστηκαν παραπάνω,
- η συνάρτηση πρέπει να επιστρέφει μια συμβολοσειρά μήκους n . Για κάθε i ($0 \leq i \leq n - 1$) ο χαρακτήρας i της συμβολοσειράς εξόδου πρέπει να είναι:
 - `'X'`, εάν το κελί (cell) i είναι μαύρο σε κάθε έγκυρη λύση,
 - `'_'`, εάν το κελί (cell) i είναι λευκό για κάθε έγκυρη λύση,
 - `'?'`, σε διαφορετική περίπτωση (π.χ., εάν υπάρχουν δύο έγκυρες λύσεις τέτοιες που το κελί (cell) i είναι μαύρο σε μία από αυτές και λευκό στην άλλη).

Στη γλώσσα C η επιγραφή της συνάρτησης είναι λιγάκι διαφορετική:

- `void solve_puzzle(int n, char* s, int k, int* c, char* result)`
 - **n**: μήκος της συμβολοσειράς **s** (πλήθος των κελιών-cells),
 - **k**: μέγεθος του πίνακα **c** (πλήθος των ενδείξεων-clues),
 - οι υπόλοιπες παράμετροι είναι ίδιες όπως παραπάνω,
 - η συνάρτηση αντί να επιστρέφει μία συμβολοσειρά από n χαρακτήρες, πρέπει να γράφει την απάντηση στην συμβολοσειρά **result**.

Οι ASCII κώδικες των χαρακτήρων που χρησιμοποιούνται σε αυτό το πρόβλημα είναι:

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

Παρακαλούμε χρησιμοποιήστε το πρότυπο αρχείων που σας παρέχεται για τις λεπτομέρειες υλοποίησης στην γλώσσα προγραμματισμού σας.

Παραδείγματα

Παράδειγμα 1

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

Αυτές είναι όλες οι πιθανές έγκυρες λύσεις του παζλ:

- `"XXX_XXXX_"`,
- `"XXX__XXXX_"`,
- `"XXX__XXXX"`,
- `"_XXX_XXXX_"`,
- `"_XXX__XXXX"`,
- `"__XXX_XXXX"`.

Παρατηρήστε ότι τα κελιά-cells (ξεκινώντας από το 0) για τις θέσεις 2, 6, και 7 είναι μαύρα σε κάθε έγκυρη λύση. Κάθε ένα από τα υπόλοιπα κελιά, μπορεί να είναι μαύρο, χωρίς αυτό να είναι απαραίτητο. Έτσι, η σωστή απάντηση είναι `"??X??X??"`.

Παράδειγμα 2

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

Στο παράδειγμα αυτό ολόκληρη η λύση προσδιορίζεται μοναδικά και η σωστή απάντηση είναι `"XXX_XXXX"`.

Παράδειγμα 3

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

Στο παράδειγμα αυτό μπορούμε να συμπεράνουμε ότι το κελί (cell) 4 πρέπει να είναι επίσης λευκό — δεν υπάρχει τρόπος να χωρέσουν τρία συνεχόμενα μαύρα κελιά μεταξύ των λευκών κελιών (cell) στις θέσεις 3 και 5. Έτσι, η σωστή απάντηση είναι `"??_???"`.

Παράδειγμα 4

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

Υπάρχουν μόνο δύο έγκυρες λύσεις που ταιριάζουν στην παραπάνω περιγραφή:

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

Έτσι, η σωστή απάντηση είναι `"?XX?_____"`.

Υποπροβλήματα

Σε όλα τα υποπροβλήματα ισχύει $1 \leq k \leq n$, και $1 \leq c_i \leq n$ για κάθε $0 \leq i \leq k - 1$.

1. (7 βαθμοί) $n \leq 20$, $k = 1$, η s περιέχει μόνο `'.'` (κενό παζλ),
2. (3 βαθμοί) $n \leq 20$, η s περιέχει μόνο `'.'`,
3. (22 βαθμοί) $n \leq 100$, η s περιέχει μόνο `'.'`,
4. (27 βαθμοί) $n \leq 100$, η s περιέχει μόνο `'.'` και `'_'` (πληροφορίες μόνο για τα λευκά κελιά),
5. (21 βαθμοί) $n \leq 100$,
6. (10 βαθμοί) $n \leq 5\,000$, $k \leq 100$,
7. (10 βαθμοί) $n \leq 200\,000$, $k \leq 100$.

Υπόδειγμα βαθμολογητή

Το υπόδειγμα βαθμολογητή διαβάζει την είσοδο στην παρακάτω μορφή:

- γραμμή 1: συμβολοσειρά s ,
- γραμμή 2: ακέραιος k ακολουθούμενος από k ακεραίους c_0, \dots, c_{k-1} .