



## რიცხვების მიხედვით ხატვა

რიცხვების მიხედვით ხატვა კარგად ცნობილი თავსატეხი თამაშია. ჩვენ განვიხილავთ ამ თავსატეხის მარტივ, ერთგანზომილებიან ვერსიას. მასში მოთამაშეს ეძლევა  $n$  რაოდენობის უჯრისაგან შედგენილი სტრიქონი. უჯრები გადანომრილია მარცხნიდან მარჯვნივ 0-დან  $(n - 1)$ -მდე. მოთამაშემ უნდა გააფერადოს თითოეული უჯრა ან თეთრად, ან შავად. 'X'-ით ჩვენ აღვნიშნავთ შავ უჯრას, ხოლო '\_' -ით კი - თეთრ უჯრას.

მოთამაშეს ასევე ეძლევა  $k$  რაოდენობის მთელი დადებითი რიცხვისაგან შედგენილი

$c = [c_0, \dots, c_{k-1}]$  მიმდევრობა: ე.წ. *მინიშნება* (ინსტრუქცია). მან უნდა გააფერადოს უჯრები ისე, რომ შავი უჯრები მოცემულ სტრიქონში ქმნიდნენ ზუსტად  $k$  რაოდენობის დამოუკიდებელ ბლოკს (თითოეული ასეთი ბლოკი მიმდევრობით განლაგებული უჯრებისაგან შედგება). გარდა ამისა, შავი უჯრების რაოდენობა მარცხნიდან  $i$ -ურ ბლოკში (დაწყებული 0-დან)  $c_i$ -ის ტოლი უნდა იყოს. მაგალითად, თუ მინიშნება არის  $c = [3, 4]$ , მაშინ ამოხსნილი თავსატეხი უნდა შედგებოდეს მიმდევრობით განლაგებული შავი უჯრებისაგან შედგენილი ზუსტად ორი ბლოკისაგან: მათგან პირველი უნდა იყოს სიგრძით 3, ხოლო მეორე სიგრძით 4. შესაბამისად, თუ  $n = 10$  და  $c = [3, 4]$ , ერთ-ერთი ამონახსნი, რომელიც აკმაყოფილებს მინიშნებას, არის: "XXX\_\_XXXX". შევნიშნოთ, რომ "XXXX\_\_XXX" არ აკმაყოფილებს მინიშნებას, რადგან შავი უჯრებისაგან შედგენილი ბლოკები არასწორი მიმდევრობითაა განლაგებული. ასევე, არც "XXXXXXXX" არ აკმაყოფილებს მინიშნებას, რადგან გვაქვს შავი უჯრებისაგან შედგენილი ერთადერთი ბლოკი და არა ორი დამოუკიდებელი ბლოკი.

თქვენ გეძლევათ რიცხვების მიხედვით ხატვის ნაწილობრივ ამოხსნილი თავსატეხი. ანუ, მოცემული გაქვთ  $n$  და  $c$ , და ასევე იცით, რომ ზოგიერთი უჯრები უნდა იყოს შავი და ზოგიერთი კი - თეთრი. თქვენი ამოცანაა დაადგინოთ და გამოიტანოთ დამატებითი ინფორმაცია უჯრების შესახებ.

კერძოდ, *სწორი ამონახსნი* არის ისეთი ამონახსნი, რომელიც აკმაყოფილებს მინიშნებას და, ამასთანავე, ეთანხმება ცნობილი უჯრების ფერებს. თქვენმა პროგრამამ უნდა იპოვოს, თუ რომელი უჯრებია გაფერადებული შავად და რომელი თეთრად ყოველ სწორ ამონახსნში.

შეგიძლიათ ჩათვალოთ, რომ ყოველი შეტანისათვის არსებობს ერთი მაინც სწორი ამონახსნი.

### იმპლემენტაციის დეტალები

თქვენ უნდა მოახდინოთ შემდეგი ფუნქციის იმპლემენტაცია (მეთოდი):

- `string solve_puzzle(string s, int[] c).`

- **s**:  $n$  სიგრძის სტრიქონი. ყოველი  $i$ -სათვის ( $0 \leq i \leq n - 1$ )  $i$ -ური სიმბოლო არის:
  - 'X', თუ  $i$ -ური უჯრა შავი უნდა იყოს,
  - '\_', თუ  $i$ -ური უჯრა თეთრი უნდა იყოს,
  - '.', თუ  $i$ -ური უჯრისათვის ინფორმაცია არ არსებობს.
- **c**:  $k$  სიგრძის მასივი, რომელიც შეიცავს მინიშნებას, ისე, როგორც ეს ზემოთაა აღწერილი,
- ფუნქციამ უნდა დააბრუნოს  $n$  სიგრძის სტრიქონი. ყოველი  $i$ -სათვის ( $0 \leq i \leq n - 1$ ) გამოსატანი სტრიქონის  $i$ -ური სიმბოლო უნდა იყოს:
  - 'X', თუ  $i$ -ური უჯრა შავია ყოველ სწორ ამონახსნში,
  - '\_', თუ  $i$ -ური უჯრა თეთრია ყოველ სწორ ამონახსნში,
  - '?', სხვა შემთხვევაში (ანუ, თუ არსებობს ორი ისეთი სწორი ამონახსნი, რომ  $i$ -ური უჯრა ერთში თეთრია, ხოლო მეორეში კი - შავი.

In the C language the function signature is a bit different:

- `void solve_puzzle(int n, char* s, int k, int* c, char* result)`
  - **n**: length of the string **s** (number of cells),
  - **k**: length of the array **c** (number of clues),
  - the other parameters are the same as above,
  - instead of returning a string of  $n$  characters, the function should write the answer to the string **result**.

ამ ამოცანაში გამოყენებული სიმბოლოების ASCII კოდებია:

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

გთხოვთ, გამოიყენოთ წარმოდგენილი შაბლონი ფაილები იმპლემენტაციის დეტალებისათვის თქვენს მიერ გამოყენებულ დაპროგრამების ენაში.

## მაგალითები

### მაგალითი 1

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

ქვემოთ მოცემულია თავსატეხის ყველა შესაძლო სწორი ამონახსნი:

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

ადვილი შესამჩნევია, რომ უჯრები ინდექსებით (დაწყებული 0-დან) 2, 6 და 7 თითოეულ სწორ ამონახსნში შავია. სხვა უჯრებიდან, ყოველი მათგანი შეიძლება იყოს შავი და შეიძლება არა. შესაბამისად, სწორი პასუხია "??X??X??".

## მაგალიტი 2

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

ამ მაგალიტში სწორი ამონახსნი ცალსახაა და, ამიტომ, სწორი პასუხია "XXX\_XXXX".

## მაგალიტი 3

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

ამ მაგალიტში ჩვენ შეგვიძლია დავასკვნათ, რომ უჯრა ინდექსით 4 თეთრი უნდა იყოს, რადგან შეუძლებელია არსებობდეს მიმდევრობით ალებული სამი შავი უჯრა ორ თეთრ უჯრას შორის ინდექსებით 3 და 5. შესაბამისად, სწორი პასუხია "??\_??\_???".

## მაგალიტი 4

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

აქ არსებობს მხოლოდ ორი სწორი ამონახსნი, რომლებიც შეესაბამება ზემოთ მოცემულ აღწერას:

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

ამგვარად, სწორი პასუხია "?XX?\_\_\_\_\_".

## ქვეამოცანები

ყველა ქვეამოცანაში  $1 \leq k \leq n$  და  $1 \leq c_i \leq n$  თითოეული ( $0 \leq i \leq k-1$ )-თვის.

1. (7 ქულა)  $n \leq 20$ ,  $k = 1$ ,  $s$  შეიცავს მხოლოდ '.'-ს (ცარიელი თავსატეხი),
2. (3 ქულა)  $n \leq 20$ ,  $s$  შეიცავს მხოლოდ '.'-ს,
3. (22 ქულა)  $n \leq 100$ ,  $s$  შეიცავს მხოლოდ '.'-ს,
4. (27 ქულა)  $n \leq 100$ ,  $s$  შეიცავს მხოლოდ '.'-ს და '\_'-ს (ინფორმაცია მხოლოდ თეთრ უჯრებზე),
5. (21 ქულა)  $n \leq 100$ ,
6. (10 ქულა)  $n \leq 5\,000$ ,  $k \leq 100$ ,
7. (10 ქულა)  $n \leq 200\,000$ ,  $k \leq 100$ .

## სანიმუშო გრადერი

სანიმუშო გრადერი შეტანას კითხულობს შემდეგ ფორმატში:

- ხაზი 1: სტრიქონი  $s$ ,
- ხაზი 2: მთელი  $k$  და შემდეგ  $k$  რაოდენობის მთელი რიცხვი  $c_0, \dots, c_{k-1}$ .