

ამოცანა 1

შექმენით ფუნქცია რომელსაც გადაეცემა String და რომელიც აბრუნებს ამ String - ში არსებულ უმოკლეს სიგეყის(სიგეყეების) სიგრძეს.

მაგალითად:

“bitcoin take over the world maybe who knows perhaps” => 3

“turns out random test cases are easier than writing out basic ones” => 3

ამოცანა 2

შექმენით ფუნქცია რომელსაც გადაეცემა მასივი და რომელიც დააბრუნებს ამ მასივში და მისი ელემენტების ქვემასივებში არსებული ყველა ელემენტის რაოდენობას.

მაგალითად:

```
([1, 2, 3]);
```

```
//>>>>> 3
```

```
(["x", "y", ["z"]]);
```

```
//>>>>> 4
```

```
([1, 2, [3, 4, [5]]]);
```

```
//>>>>> 7
```

ამოცანა 3

შექმენით ფუნქცია რომელიც გადმოცემულ String - ში შეცვლის ყველა სიმბოლოს შემდეგი პრინციპით

- სიმბოლოები რომლებიც გვხვდება მხოლოდ ერთხელ იცვლება “(” - ით
- სიმბოლოები რომლებიც გვხვდება 1 - ზე მეტჯერ იცვლება “)” - ით.

მაგალითად:

"din" => "(((

"recede" => "()()())"

"Success" => ")()())"

"((@" => "))(

შენიშვნა : არ მიქცეით ყურადღება ღიდ და პატარა ასოებს.

ამოცანა 4

შექმენით ფუნქცია რომელიც გადმოცემულ String - ის ექსპანსიას ახდენს შემდეგი პრინციპით:

`solve("3(ab)") = "ababab" -- "ab" მეორდება 3 ჯერ`

`solve("2(a3(b))") = "abbbabbb" -- "a3(b)" == "abbb"`
მეორდება ორჯერ.

String - ში გადმოგეცემათ მხოლოდ დაბალი რეგისტრის ასოები და რიცხვები 1 - 9 - ის ჩათვლით. ფრჩხილები ყოველთვის იქნება დასმული ვალიდურად.

ამოცანა 5

დაწერეთ ფუნქცია `rangeExtraction(array)` რომელსაც გადაეცემა ერთგანზომილებიანი მასივი შედგენილი რიცხვებისგან (უარყოფითი რიცხვები, ნული და დადებითი რიცხვები) და რომელიც დააბრუნებს String - ს შემდეგი პრინციპით

- თუ მასივში გვხვდება რიცხვების მრავალი მიმდევრობა რომლის წევრთა რაოდენობა მეტია ან ტოლია სამზე ვაბრუნებთ შემდეგი სახით : a-b. მაგ. (-6,-5,-4,-3,-2) გვაძლევს -6--2
- ისეთი რიცხვები რომლებიც არ არიან მიმდევრობის წევრები ბრუნდებიან ინდივიდუალურად.

`rangeExtraction([-6,-3,-2,-1,0,1,3,4,5,7,8,9,10,11,14,15,17,18,19,20])`

უნდა დააბრუნოს შემდეგი String:

`"-6,-3-1,3-5,7-11,14,15,17-20"`

ამოცანა 6

დაწერეთ ფუნქცია `snail(array)` რომელსაც გადაეცემა $n \times n$ მასივი და რომელიც დააბრუნებს მასივის ელემენტებს, დალაგებულს ნულოვანი ელემენტიდან მასივის ცენტრალურ ელემენტამდე საათის ისრის მიმართულებით შემორტყმული ბრუნის მიხედვით.

1	2	3
4	5	6
7	8	9

1	2	3	1
4	5	6	4
7	8	9	7
7	8	9	7

მაგალითად:

მასივისთვის [[1,2,3],
[4,5,6],
[7,8,9]]

ფუნქციამ უნდა დააბრუნოს [1,2,3,6,9,8,7,4,5]

ხოლო მასივისთვის [[1,2,3],
[8,9,4],
[7,6,5]]

ფუნქციამ უნდა დააბრუნოს [1,2,3,4,5,6,7,8,9].