

# Задачи за напреднали

11 юли 2018 г. 13:14

## Структури от данни:

1. Да се имплементира:
  - a. Stack от Number (Java има вградена структура LinkedList, която е удобна за имплементацията)
    - i. Методи :
      - 1) void push(Number),
      - 2) Number pop(),
      - 3) Number peek(),
      - 4) boolean isEmpty()
  - b. Queue от Number (Java има вградена структура LinkedList, която е удобна за имплементацията)
    - i. Методи:
      - 1) void enqueue(Number),
      - 2) Number dequeue(),
      - 3) boolean isEmpty()
  - c. Linked List от Number (Без да се използва вградената структура в Java)
    - i. Методи:
      - 1) void addLast(Number),
      - 2) void addFirst(number),
      - 3) Number removeLast(),
      - 4) Number removeFirst(),
      - 5) Number removeAtPosition(int)
  - d. Graph (С използването на списък на съседство (Adjacency list))
    - i. Методи:
      - 1) addVertex(int)
      - 2) addEdge(int,int)
      - 3) removeVertex(int)
      - 4) removeEdge(int)

## Design Patterns:

1. Singleton
  - a. Да се създаде клас Connection който ни дава достъп до някакъв уебсайт чрез неговото URL (String) - връзката е теоретична.
  - b. Класа да има методи String getConnection(), който връща съобщение че връзката е установена за определения URL и метод Connection getInstance(), който връща инстанцията според Singleton pattern-а
2. Factory Method
  - a. Да се създаде интерфейс Food, който има методи String getFood(), който връща името на храната и String getPriceIn(Currency), който връща цената като String в валутата, която е му е подадена.
    - i. Currency е клас в Java (<https://docs.oracle.com/javase/8/docs/api/java/util/Currency.html>)
  - b. Да се създадат класове Pizza, Cake, FruitBowl които имплементират Food интерфейса.
  - c. Да се създаде клас FoodFactory
    - i. С метод Food getOrder(String), който връща обекти от различните класове храни

## Алгоритми:

### Left Rotation

A left rotation operation on an array shifts each of the array's elements unit to the left. For example, if left rotations are performed on array , then the array would become .

Given an array of integers and a number, , perform left rotations on the array. Return the updated array to be printed as a single line of space-separated integers.

#### Function Description

Complete the function `rotLeft` in the editor below. It should return the resulting array of integers.

`rotLeft` has the following parameter(s):

- An array of integers .
- An integer , the number of rotations.

#### Input Format

The first line contains two space-separated integers and , the size of and the number of left rotations you must perform.

The second line contains space-separated integers .

#### Constraints

#### Output Format

Print a single line of space-separated integers denoting the final state of the array after performing left rotations.

#### Sample Input

```
5 4
```

```
1 2 3 4 5
```

#### Sample Output

```
5 1 2 3 4
```

#### Explanation

When we perform left rotations, the array undergoes the following sequence of changes:

$$[1, 2, 3, 4, 5] \rightarrow [2, 3, 4, 5, 1] \rightarrow [3, 4, 5, 1, 2] \rightarrow [4, 5, 1, 2, 3] \rightarrow [5, 1, 2, 3, 4]$$

### Super reduced string

Steve has a string of lowercase characters in range `ascii['a'..'z']`. He wants to reduce the string to its shortest length by doing a series of operations in which he selects a pair of adjacent lowercase letters that match, and then he deletes them. For instance, the string `aab` could be shortened to `b` in one operation.

Steve's task is to delete as many characters as possible using this method and print the resulting string. If the final string is empty, print `Empty String`

#### Input Format

A single string, .

#### Constraints

#### Output Format

If the final string is empty, print `Empty String`; otherwise, print the final non-reducible string.

#### Sample Input 0

```
aaabccddd
```

#### Sample Output 0

```
abd
```

#### Explanation 0

#### Sample Input

```
aaabccddd
```

#### Sample Output 0

```
abd
```

#### Explanation

Steve performs the following sequence of operations to get the final string:  
aaabccddd → abccddd → abddd → abd

Sample Input 1

aa

Sample Output 1

Empty String

Explanation 1

Sample Input 1

aa

Sample Output 1

Empty String

Explanation 1

aa → Empty String

Sample Input 2

baab

Sample Output 2

Empty String

Explanation 2

Sample Input 1

baab

Sample Output 1

Empty String

Explanation 1

baab → bb → Empty String

### Alternating characters

You are given a string containing characters `a` and `b` only. Your task is to change it into a string such that there are no matching adjacent characters. To do this, you are allowed to delete zero or more characters in the string.

Your task is to find the minimum number of required deletions.

For example, given the string `aabab`, remove an `a` at positions `0` and `2` to make `bab` in `3` deletions.

### Function Description

Complete the `alternatingCharacters` function in the editor below. It must return an integer representing the minimum number of deletions to make the alternating string.

`alternatingCharacters` has the following parameter(s):

`s`: a string

Input Format

The first line contains an integer `t`, the number of test cases.

The next `t` lines contain a string `s`.

### Constraints

Each string `s` will consist only of characters `a` and `b`.

Output Format

For each test case, print the minimum number of deletions required on a new line.

Sample Input

5  
AAAA  
BBBBB  
ABABABAB  
BABABA  
AAABBB

Sample Output

3  
4  
0  
0  
4

Explanation

The characters marked red are the ones that can be deleted so that the string doesn't have matching consecutive characters.

**AAAA** -> A (3 deletions)

**BBBBB** -> B (4 deletions)

ABABABAB -> ABABABAB (0 deletions)

BABABA -> BABABA (0 deletions)

**AAABBB** -> AB (4 deletions)

## Java:

### Регулярни изрази

Да се използват регулярни изрази (regex) за да се провери дали в определен текст се съдържат числа от вида 5.6, 3.0, 1.5 и да ги изведе на конзолата

Примерен вход:

"Hello I would like to get 5 apple for 1.5 leva and 2 pumpkins for 3.4 leva."

Изход:

1.5  
3.4

Полезни методи са :

В класа Matcher boolean find() и String group(int)

Може да прочетете за регулярни изрази и как се използват в Java тук:

<https://docs.oracle.com/javase/8/docs/api/java/util/regex/Pattern.html>

<https://regexr.com/>

<https://docs.oracle.com/javase/tutorial/essential/regex/>

### BigInteger

Да се използва класа BigInteger за да се имплементират два метода

1. Метод, който изчислява факториел на големи числа
2. Метод, който намира редицата на Фибоначи с голям на брой елементи

## Multi threading

Да се разгледа механизма на Java за много нишкови приложения и да се създаде следната програма:

Две нишки, които принтират числата от 1 до 10, едно след друго.

<https://www.journaldev.com/1016/java-thread-example>

Синхронизацията може да се извърши с `sleep()`

## Lambda expressions

Разгледайте Lambda expressions и дайте примери за това как могат да се използват.

<https://docs.oracle.com/javase/tutorial/java/javaOO/lambdaexpressions.html>