

## Algorithmen und Datenstrukturen

Exercise sheet 4

---

### 1 Lonely Element

Implementieren Sie eine Methode `getLonelyElement` welche als Parameter einen `string` bekommt. Jeder Character in diesem `string` kommt zweimal vor, ausser eines. Die Methode soll das Element zurückliefern, welches nur einmal vorkommt.

```
1 class StringUtil {  
2 public:  
3     static char getLonelyElement(string input);  
4 };
```

Beispiel:

Input: abcdedcbal l

Resultat: e (das Element e kommt nur einmal vor)

**ACHTUNG:** Die Implementierung muss besser sein als  $O(n^2)$

### 2 Plus Minus

Implementieren Sie eine Methode `analyseArray` welche als Parameter einen `vector` mit Integer Werten bekommt. Diese Werte sind entweder Positiv, Negativ oder Null. Die Methode soll nun berechnen, wie viel Prozent der Werte Positiv, Negativ resp. Null sind. Diese drei Werte werden als Objekt der Klasse `Result` zurückgeliefert.

```
1 class Result {  
2 public:  
3     int nPositive;  
4     int nZero;  
5     int nNegative;  
6 };  
7  
8 class ArrayUtil {  
9 public:  
10     static Result analyseArray(vector<int> input);  
11 };
```

Beispiel:

Input: 1, 10, 0, -6, -9, 2, 5

Lösung: [nPositive: 57.1%; nZero: 14.3%; nNegative: 28.6%]