

Algorithmen und Datenstrukturen

Exercise sheet 1

1 Analyse

Welche Laufzeit hat der folgende Code?

```
1 void foo(int *array, int size) {  
2     int sum = 0;  
3     int product = 1;  
4     for (int i=0; i<size; i++) {  
5         sum += array[i];  
6     }  
7     for (int i=0; i<size; i++) {  
8         product *= array[i];  
9     }  
10    cout << sum << ", " << product << endl;  
11 }  
alc
```

2 Array Multiplikation

In dieser Aufgabe soll eine Methode implementiert werden, welche als Parameter einen Array enthält. Das Resultat ist ein Array der gleichen Grösse. Jedes Element dieses Arrays beinhaltet die Multiplikation aller Elemente (ausser sich selbst) des Input Arrays.

Beispiel:

Input Array: {12, 4, 7, 3}

Output Array: {4*7*3, 12*7*3, 12*4*3, 12*4*7} = {84, 252, 144, 336}

ACHTUNG: Der Divisionsoperator darf in dieser Aufgabe NICHT verwendet werden!

```
1 class ArrayUtil {  
2 public:  
3     static long* multiplyArrayValues(const long *values, const int ARRAY_SIZE);  
4 };
```

3 String Check

In dieser Aufgabe soll eine Methode implementiert werden, welche als Parameter einen String enthält. Das Resultat ist ein Boolean. `true` falls alle Buchstaben unique sind (alle Buchstaben kommen nur einmal vor), sonst `false`.

```
1 class StringUtil {  
2 public:  
3     static bool checkUniqueness(string s);  
4 };
```