

Algorithmen und Datenstrukturen

Exercise sheet 3

1 Münzkombinationen

In dieser Aufgabe soll ein Algorithmus implementiert werden, welcher für einen bestimmten Geldbetrag die minimale Anzahl Münzen berechnet.

Der Algorithmus erhält als Input den Betrag sowie ein Array mit den verfügbaren Münzen.

```
1 class CoinCombination {  
2 public:  
3     static void printCoinCombination(  
4         double value, // Betrag  
5         const double *coins, // Pointer auf Muenzarray  
6         const int ARRAY_SIZE); // Groesse des Muenzarray  
7 };
```

Beispiel:

Betrag: 23 CHF

Münzen: 0.05, 0.10, 0.20, 0.50, 1, 2, 5

Resultat: 5, 5, 5, 5, 2, 1 (minimale Anzahl Münzen)

2 Palindrome

In dieser Aufgabe soll eine Methode implementiert werden, welche mit Hilfe von Rekursion prüft, ob eine Zeichenkette ein Palindrome ist.

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

3 Längster gemeinsamer Substring

In dieser Aufgabe soll eine Methode implementiert werden, welche von 2 Zeichenketten den längsten Substring findet, welcher in beiden Zeichenketten vorkommt.

```
1 class StringUtil {  
2 public:  
3     static string lcs(string input1, string input2);  
4 };
```