**Øvelser stak og kø**

**Øvelse 1: Metode som kan tjekke matchende parenteser i en string**

**Mål**: Skriv en metode, som returnerer true eller false, og som har en string som parameter. Opgaven går ud på at tjekke, om teksten indeholder matchende parenteser. Hvis den gør, returneres true; hvis ikke returneres false.

For at løse opgaven korrekt er det et krav, at der anvendes en stak. Ideen er, at når man møder en venstreparentes, pushes et element på stakken, og når man møder en højreparentes, skal der poppes. Hvis der undervejs forsøges at poppe på en tom stak, returneres false. Hvis stakken er tom, når hele teksten er læst, returneres true; hvis den ikke er tom, returneres false.

Løsningen skal tage højde for at der kan forekomme andet i teksten end start- og slutparenteser.

**Øvelse 2: Klasse som kan repræsentere en kø**

**Mål**: En stak opererer med LIFO (last in, first out). For en kø er princippet FIFO, og opgaven går ud på at skrive en template klasse ved navn Queue, som kan repræsentere en kø.

Metoden, som tilføjer elementer til køen, skal hedde *enqueue,* og metoden, som fjerner elementer fra køen, hedder *dequeue.*

Til sidst foretages en passende test af klassen.

**Exercises templates**

**Exercise 1: Template Function to Find Maximum Element in an Array**

**Goal**: Create a function template findMax that returns the maximum value in a C-style array.

**Exercise 2: Template Class for a Simple Pair**

**Goal**: Implement a template class Pair that holds two values (like std::pair).

**Exercise 3: Template Function to Reverse an Array**

**Goal**: Write a function template that reverses the elements in a C-style array.