

Hausaufgabe 1.3: Maximum-Subarray-Problem

```
import java.lang.*;
import java.math.*;
import java.io.*;

public class Subs
{
    public static int input[] = {12, -32, 38, 22, -27, 32, 17, -13, -10, 10};

    public static void main(String[] args)
    {
        //Initialisierung
        int start = 0;
        int maxSum = Integer.MIN_VALUE;
        int maxStart = 0;
        int maxEnd = 0;
        int aktSum = 0;

        //Checkpoint init
        for(int i = 0; i < input.length; i++)
        {
            //Berechnen der aktuellen Summe
            aktSum += input[i];
            //Checkpoint (1)
            //Ist Teilarray positiv?
            if(aktSum > 0)
            {
                //Erweiterung des Teilarrays sinnvoll
                if(aktSum > maxSum)
                {
                    //Merken des neuen besten Teilarrays
                    maxSum = aktSum;
                    maxStart = start;
                    maxEnd = i;
                }
            }

            else // Teilarray mit Summe kleiner 0 => Beginne neues Array bei nächster Position
            {
                //Falls maxSum negativ kann input[i] immer noch größer sein
                if(input[i] > maxSum)
                {
                    maxSum = input[i];
                    maxStart = i;
                    maxEnd = i;
                }

                start = i + 1;
                aktSum = 0;
            }

            // Checkpoint (2)
        }

        System.out.println("TF: " + maxStart + " bis " + maxEnd + " mit Summe : " + maxSum + " ist maximal !");
    }
}
```