Offline test

Please answer as many questions as you can. Feel free to answer in any order that you feel comfortable with.

1. **Coding task 1. Longest substring without repeating characters.**

   Given a string, find the length of the **longest substring** without repeating characters.

   Example 1:
   Input: "abcabcbb"
   Output: 3
   Explanation: The answer is "abc", with the length of 3.

   Example 2:
   Input: "bbbbb"
   Output: 1
   Explanation: The answer is "b", with the length of 1.

   Example 3:
   Input: "pwwkew"
   Output: 3
   Explanation: The answer is "wke", with the length of 3.

2. Coding task 2. Find subarray with given sum.

Given an unsorted array of nonnegative integers. Write a function which returns **`true`** if there is a **continuous** subarray which adds to a given number (target sum). Estimate the complexity of your code.

Example:
Input: [1, 2, 3, 4] and target sum is 7
Output: true
Explanation: Sum of subarray [3, 4] is 7

3. Java: List all issues you can find in the following class:

```
import java.util.HashMap;
import java.util.Map;

import org.slf4j.LoggerFactory;

public abstract class Digest {

    private Map<byte[], byte[]> cache = new HashMap<byte[], byte[]>();

    public byte[] digest(byte[] input) {
        byte[] result = cache.get(input);
        if (result == null) {
            synchronized (cache) {
                result = cache.get(input);
                if (result == null) {
                    try {
                        result = doDigest(input);
                        cache.put(input, result);
                    } catch (RuntimeException ex) {
                        LoggerFactory.getLogger("Digest").error(
                            "Unable to make digest"
                        );
                        throw ex;
                    }
                }
            }
        }
        return result;
    }

    protected abstract byte[] doDigest(byte[] input);
}
```