Lab 1: SHA-256 Hashing
This lab implements SHA-256 hashing algorithm in Java.

Author
Name: Oleksii Shyshyma

Group: KH-M922в

Task
The goal of the lab is to implement SHA-256 hashing algorithm in Java.

Solution
The SHA-256 algorithm consists of several rounds of hashing, where each round operates on a block of 512 bits. In each round, a specific set of constants and working variables are used to update the hash value. The algorithm operates on a message, and the message is padded to ensure it is a multiple of 512 bits long.

The code consists of a Sha256 class that provides a hash(byte[] message) method for hashing messages with SHA-256. It uses the following constants and working variables:

K: a constant array of length 64
H0: a constant array of length 8
BLOCK_BITS: the size of a block in bits (512)
BLOCK_BYTES: the size of a block in bytes (64)
W: a working array of length 64
H: a working array of length 8
TEMP: a working array of length 8
The hash method uses these constants and variables to hash the message.

The implementation follows the SHA-256 algorithm as described in the NIST FIPS 180-4 publication.

Results
The implementation has been tested and works correctly.

Example usage:

java
Copy code

```
byte[] message = "Hello, world!".getBytes();
byte[] hash = Sha256.hash(message);
System.out.println("Hash: " + bytesToHex(hash));
```

Output:

makefile
Copy code
Hash: a591a6d40bf420404a011733cfb7b190d62c65bf0bcda32b57b277d9ad9f146e