CT255 Assignment 1

Breaking Hash Functions

General

The objectives of this assignment are as follows:

1. Reinforce your understanding of hash functions.
2. Implement and apply brute-force concepts to find hash collisions.

The Java class *CT255_HashFunction1* (please see Blackboard attachment) is a quick and dirty hash function that translates a String (1 to 64 characters) into a 32-bit hash value (which is for the sake of simplicity returned as a positive 32 bit integer value). The hash code is by no means bullet proof, but nonetheless forms the basis of this assignment. Just import or copy&paste the code into your Java IDE.

In detail, the class *CT255_HashFunction1* consists of a

- main() function that takes in a command line argument (i.e. the input value)
- hash function *hashF1()* that does all the donkey work.

Problem 1: [2 marks]
Study the code and summarize its functionality (bullet points will do), thereby referencing important lines of source code.

Problem 2: [4 marks]
Consider the input "Bamb0" (i.e. "Bamb" followed by a zero). The resulting hash value is 1079524045.
Enhance the code to search for "Bamb0" hash collisions (i.e., different inputs that create the same hash value → weak collision resistance) via a brute-force search.
What collision(s) can you find?

Problem 3: [4 marks]
Enhance the code in *hashF1()* to make it more robust, i.e., to reduce the risk of hash collisions. Explain your answer via comments in your code.

Assignment Submission

Please submit a zipped folder to Blackboard containing:

- Your answer for problem 1.
- Your (well-commented!) source code for problems 2 and 3, all in PDF format.
- For problem 2 screenshots showing your program being compiled and producing results, i.e., a list of (up to 10) hash collisions you could identify.
- For problem 3 screenshots showing your program being compiled a summary of your enhancements embedded in your source code comments.