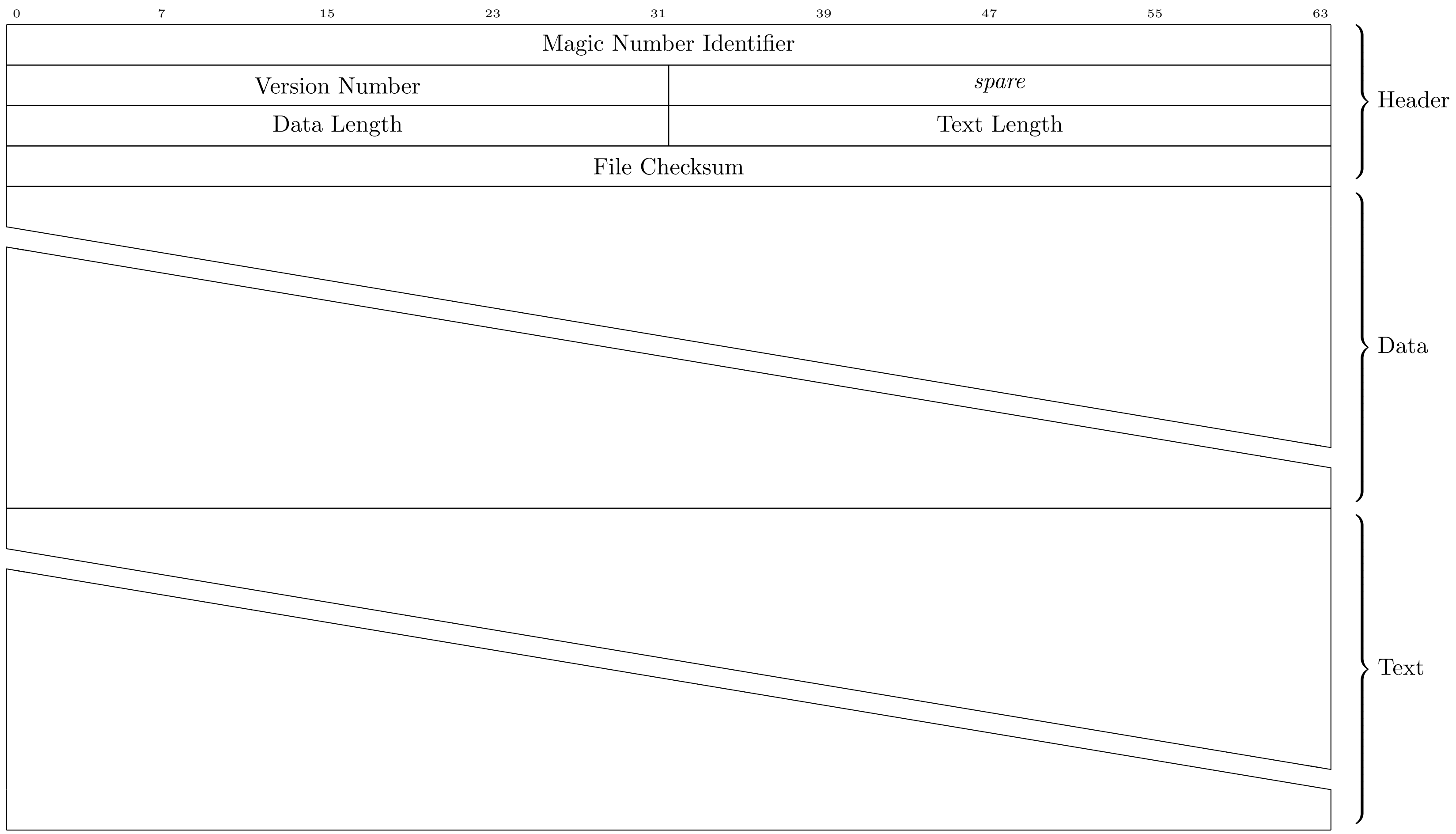
File Format


0
7
15
23
31
39
47
55
63
Magic Number Identifier
Version Number
spare
Data Length
Text Length
File Checksum
Header
Data
Text

Field descriptions

Magic Number Identifier

0xF0 0x9F 0x8D 0x8C 0xF0 0x9F 0x8D 0x8C

A unique 64-bit value to mark the beginning of a `.bb` file. Value is two copies of the UTF-8 encoding for the "Banana" emoji concatenated.

Version Number

BNA version for this file. Not currently used.

Data Length

Length of the Data segment in bytes.

Text Length

Length of the Text segment in bytes.

Data Segment

Entry Format

<table>
<tr><td>identifier</td><td><em>spare</em></td><td>type</td><td>data length</td></tr>
<tr><td colspan="4">data</td></tr>
</table>

Identifier

Identifier for this value. Used in instructions to find the value of this literal.

Type

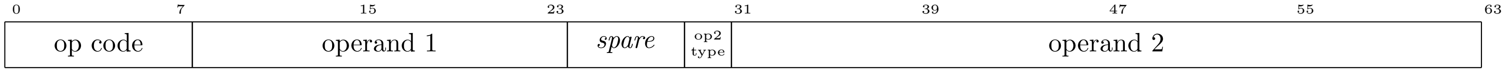
Data type of the value(s).

Length

Length of the data entry in **bytes**.

Text Segment

Instruction Format



op code

Identifier for the instruction to execute. Two special values, `0x00` exits the program and `0xFF` represents an invalid or unknown instruction (possibly will use for NOP later).

operand 1

Identifier for the first operand of the instruction. Special value `0x00` is `null` and is for instructions with no first operand.

op2 type

The type of operand in the `operand 2` field. Uses the below table for the type values.

Code	**Type**	**Value**
`0x00`	None	There is no second operand for this operation.
`0x01`	Variable	Stored as a variable, with `operand 2` as the identifier.
`0x10`	Literal	Stored as a literal, with `operand 2` as the identifier.
`0x11`	Small Literal	Value is `operand 2` as a sign-extended integer.

operand 2

Identifier or value for the second operand. See `op2 type`