Address in Method Area (hex)	Opcode stored in method area (hex)	Mnemonic (an assembler converts these into opcodes)
		.main
		.var
		i
		k
		.end-var
		START:
0000	FC	**IN**
0001	59	**DUP**
0002	36 00	**ISTORE i**
0004	10 00	**BIPUSH 0**
0006	9F FF FA	**IF_ICMPEQ START**
0009	15 00	**ILOAD i**
000B	10 30	**BIPUSH 48**
000D	64	**ISUB**
000E	10 05	**BIPUSH 5**
0010	64	**ISUB**
0011	9B 00 0A	**IFLT DO**
0014	10 08	**BIPUSH 8**
0016	36 01	**ISTORE k**
0018	A7 00 07	**GOTO SKIP**
001B	10 07	**DO: BIPUSH 7**
001D	36 01	**ISTORE k**
001F	15 01	**SKIP: ILOAD k**
0021	10 30	**BIPUSH 48**
0023	60	**IADD**
0024	FD	**OUT**
0025	10 20	**BIPUSH 0x20**
0027	FD	**OUT**
0028	A7 FF D8	**GOTO START**
		.end-main

The entire program is 43 bytes long and resides in the method area starting at address 0000 and ending at address 002A.

Address calculation for IF_ICMPEQ START

		START:
0000	FC	**IN**
0001	59	**DUP**
0002	36 00	**ISTORE i**
0004	10 00	**BIPUSH 0**
0006	9F FF FA	**IF_ICMPEQ START**

If i is equal to zero then the branch will be taken. That means the PC needs to be adjusted 6 locations backward from 0006 to 0000. This is equivalent to moving PC -6 locations.

Since relative offsets are always expressed with 16 bits in the Mic-1, let's write minus 6 in two's complement representation and then express it in hex.

write +6 in binary: 00000000 00000110 (0006H)
take two's complement: 11111111 11111010 (FFFAH)

OR simply subtract in base 16

```
  10000 H
  -0006 H
  ________
   FFFA H
```

Address calculation for GOTO SKIP

0018	A7 00 07	**GOTO SKIP**
001B	10 07	**DO: BIPUSH 7**
001D	36 01	**ISTORE k**
001F	15 01	**SKIP: ILOAD k**

The branch is unconditional and we need to advance PC from 0018 to 001F.

```
  001F H
 -0018 H
 ________
  0007 H
```