BitBitJump

Problem author and developer: Georgiy Korneev

There are a lot of different solutions. One of them contains two blocks of 17 instructions each. The first block starts from word 0, and the second one from word 64.

The i-th instruction of the block, counting from zero, depends on the $(i-1)$-th bit of the x, where bit -1 is considered to be zero.

Let $\mathrm{IO}_{\mathrm{word}} = 2^{12} - 1$ and $\mathrm{IO}_{\mathrm{bit}} = \mathrm{IO}_{\mathrm{word}} \cdot 16$ — address of the IO word and its first bit. For each instruction let $@_{\mathrm{word}}$ and $@_{\mathrm{bit}}$ to be addresses of its first word and bit.

For $i = 0..15$ if the $(i-1)$-th bit of the input is zero, then in the i-th instruction of the first is `bbj` $\mathrm{IO}_{\mathrm{bit}}+i, @_{\mathrm{bit}}+32+6, @_{\mathrm{word}}+3$, and instruction of the second block is “`return false`“: `bbj` $0, \mathrm{IO}_{\mathrm{bit}}, \mathrm{IO}_{\mathrm{word}}$. If the $(i-1)$-th bit of the input is one, then instructions in the blocks are swapped.

If the 15-th bit of x is zero, then the last instruction of the first block is “`return true`“: `bbj` $4, \mathrm{IO}_{\mathrm{bit}}, \mathrm{IO}_{\mathrm{word}}$, and in the second block is “`return false`“. Otherwise, these instructions also should be swapped.