Question 2 (30 points): There are two branch instructions in the Instruction Set Architecture of *TinyProc*. The opcode for `beq` is `010` and the opcode for `blt` is `011`. When writing the MIPS assembly code below, you cannot use pseudo-instructions that use constants that are larger than 16 bits.

1. (**10 points**) Write, in MIPS assembly, a subroutine called `IsBranch` that receives in `$a0` a memory address. If the *TinyProc* instruction at that address is a branch, then `IsBranch` returns `$v0 = 1` otherwise `IsBranch` returns `$v0 = 0`. Obey all the MIPS calling conventions.

2. (**20 points**) Write, in MIPS assembly, a subroutine called `CountBranches` that receives the address of the first instruction in a *TinyProc* program in `$a0` and returns in `$v0` the number of branches found in the program. The instructions of this *TinyProc* are stored continuously in memory and the end of the program is signalled by a half word containing `0xFFFF`. `CountBranches` must call `IsBranch` to identify if an individual instruction is a branch. It must follow all the MIPS calling conventions.

MIPS code for `IsBranch`

MIPS Code for `CountBranches`