

Question 1: (20 points)

You are participating in the Computing Science Industrial Internship Program and your placement is with *Tiny Inc.*, a company that produces *TinyProc*— a new processor developed for the automobile industry. All instructions in *TinyProc* have 16 bits. *TinyProc* also works with 16-bit addresses. The format of a branch instruction in *TinyProc* is as shown below:

15	13	12	10	9	7	6	0
Opcode		rs		rt		address	

Where **rs** and **rt** specify the source and target registers for the branch instruction, respectively. The address of the target of a branch instruction is computed using the same mechanism used in the MIPS processor, but the increment of the PC and the shift left have to be adjusted for a 16-bit address machine: first the Program Counter (PC) is incremented by **two**, then the bitfield **address** of the branch instruction is shifted left by **one**, sign-extended to sixteen bits, and added to the incremented PC.

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.

- (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.
- (10 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.