

**lw** dest, offset(base) -> đọc

**sw** source, offset(base) -> copy nội dung của source và cho vào adresse offset(base)

**li** dest, constant

**addi** dest, source, constant

**move** dest, source -> cho source vào dest

**neg** dest, source -> vd: 10 sẽ thành -10

**add** dest, source1, source2

de même pour sub, mul, div

**slt** dest, source1, source2 -> set on less than if s1 < s2 then dest = 1; else dest = 0

**bgtz** source, address -> si contenu du source > 0, saute à address(vd : **bgtz** %1, f10)

**bgez** source, address (branch on greater than or equal to zero)

**blez** source, address (branch on less than or equal to 0)

**bltz** source, address (branch on less than zero)

**beq** source1, source2, address -> branch on equal (vd: **beq** %0, %1, f10)

**blt** source1, source2, address -> branch on less than

**bne** source1, source2, address -> branch on not equal to

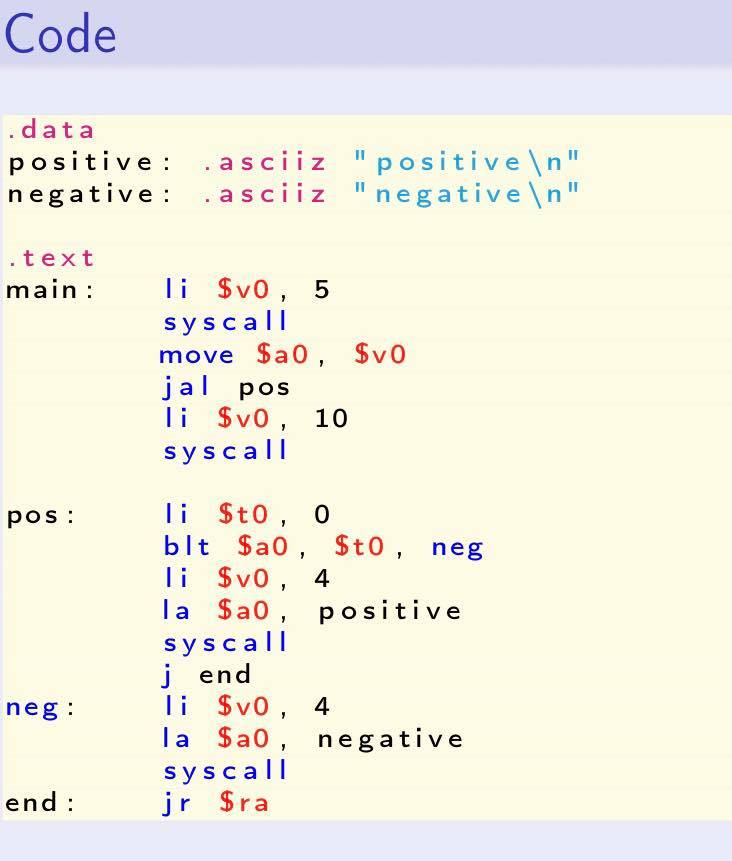
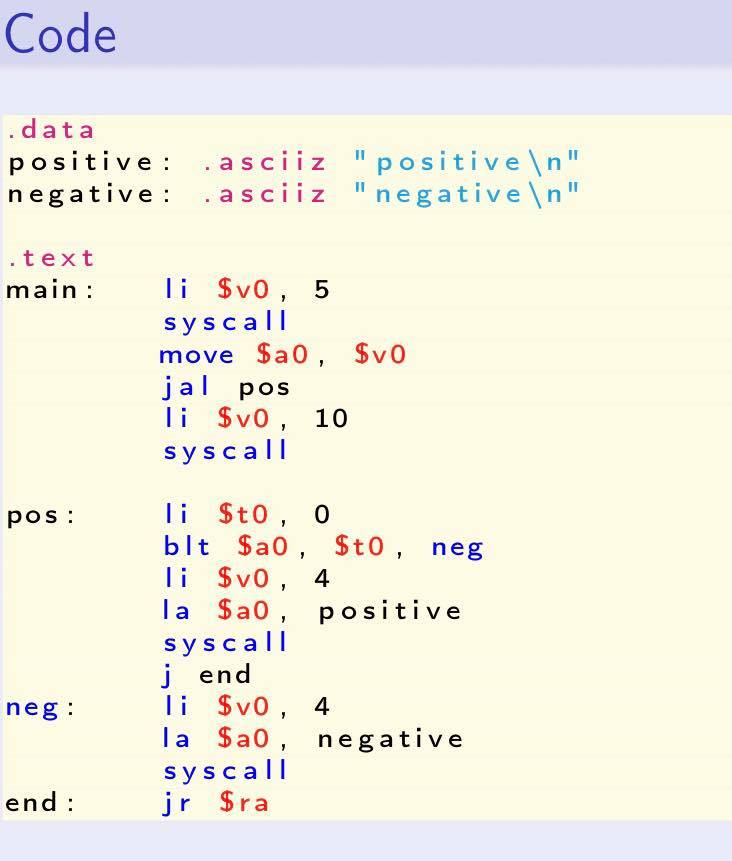
**j** address -> vd: j f10

**jal** address -> lưu adresse của instruction sau vào $ra và nhảy đến adresse “address”

**jr** target -> jump register (vd: **jr** $ra )

**syscall** -> appel du système pour afficher les arguments du register $v0 và $a0 - $a3

vd : MIPS



Ici **jal pos** lưu instruction « li $v0, 10 » vào $ra và nhảy đến **pos :** và sau khi **pos :** xong thì nhảy đến **end :** và end sẽ **jr $ra** tức là nhảy về lại instruction « li $v0, 10 »