I-type

- Branch: conditional opcode

Opcode	Operands

Operands:

2 registers, 1 label

```
Beq $t0,$t1, Next  # branch if equal
Bne $t0,$t1, Next  # branch if not equal
Bgt $t0,$t1, Next  # branch if greater than
Blt $t0,$t1, Next  # branch if lower than
```

1 register, 1 value, 1 label

```
Beq $t0, 10, Next  # if t0=10
```

….

I-type

1 register, 1 label

Beqz $t0, Next # branch if $t0 contains 0

…

J-type

Type 3: J-type (Jump instructions)

Opcode	Operands

Operand: label

Opcode:

- j End : jumps to the label End
- jal Min (used in functions) : copies the address of the next instruction into the register $ra (return address) and then jumps to Min.
- jr $ra: jumps to the address in $ra

If Then Else Statement

branch_opcode **operands** (includes **Label**)

else instructions

label:

if instructions

Example:

```
beq $t0,$t1, Next
add $t2, $t0,$t1
j End
Next:
sub $t2, $t0, $t1
End:
```

Read integer
End Program

Read an integer:

v0 🡐 5

syscall

# v0 contains the entered integer

End Program

v0 🡐 10

syscall

Exercise

Write a program to determine the maximum of 2 integers entered at the keyboard.

Write a program to determine whether an integer number is odd or even.