**COMP 122/L**
**Summer 2023**

## Branching MIPS Assembly (Answers)

For each of the following problems, translate the given C-like code to MIPS assembly. If a register is used in the C-like code, your MIPS code should use the register in the same way. You may use additional registers not used in the C-like code.

1.)

```
if ($t0 == 0) {
  $t1 = 5;
}
```

```
  bne $t0, $zero, after_if
  li $t1, 5
after_if:
```

2.)

```
if ($t0 != 0) {
  $t1 = 6;
} else {
  $t1 = 7;
}
```

```
  beq $t0, $zero, else_branch
  li $t1, 6
  j after_if
else_branch:
  li $t1, 7
after_if:
```

3.)

```
do {
  $t0 = $t0 + 1;
} while ($t0 != 12);
```

```
  li $t1, 12
loop_begin:
  addiu $t0, $t0, 1
  bne $t0, $t1, loop_begin
```

4.)

```
while ($t0 != 1000) {
  $t0 = ($t0 + 1) * 2;
}
```

```
  li $t1, 1000
  li $t2, 2
loop_begin:
  beq $t0, $t1, loop_end
  addiu $t0, $t0, 1
  multu $t0, $t1
  mflo $t0
  j loop_begin
loop_end:
```

5.)

```
if ($t0 < 10) {
  $t1 = 5;
}
```

```
  sltiu $t2, $t0, 10
  beq $t2, $zero, if_end
  li $t1, 5
if_end:
```

6.)

```
if ($t0 > 10) {
  $t1 = 3;
}
```

```
  # $t0 > 10 is equivalent to 10 < $t0
  li $t2, 10
  sltu $t3, $t2, $t0
  beq $t3, $zero, after_if
  li $t1, 3
after_if:
```

7.)

```
if ($t0 >= 10) {
  $t1 = 5;
}
```

```
  # $t0 >= 10 is equivalent to !($t0 < 10)
  sltiu $t2, $t0, 10 # $t0 < 10?
  bne $t2, $zero, after_if  # if $t0 < 10, jump to end
  li $t1, 5
after_if:
```

8.)

```
if ($t0 <= 10) {
  $t1 = 5;
}
```

```
  # $t0 <= 10 is equivalent to !($t0 > 10),
  # which is equivalent to !(10 < $t0)
  li $t2, 10
  sltu $t3, $t2, $t0 # 10 < $t0?
  bne $t3, $zero, if_end  # if 10 < $t0, jump to end
  li $t1, 5
if_end:
```

9.)

```
while ($t0 <= 5) {
  $t0 = $t0 + 1;
}
```

```
  # $t0 <= 5 is equivalent to !($t0 > 5), which is
  # equivalent to !(5 < $t0)
  li $t1, 5
loop_begin:
  sltu $t2, $t1, $t0  # 5 < $t0?
  bne $t2, $zero, loop_end  # if 5 < $t0, jump out of the loop
  addiu $t0, $t0, 1
  j loop_begin
loop_end:
```

10.)

```
while ($t0 > 1) {
  $t0 = $t0 / 2;
}
```

```
  # $t0 > 1 is equivalent to 1 < $t0
  li $t2, 1
  li $t3, 2
loop_begin:
  sltu $t1, $t2, $t0  # 1 < $t0?
  beq $t1, $zero, loop_end  # if !(1 < $t0), jump out of loop
  divu $t0, $t3
  mflo $t0
  j loop_begin
loop_end:
```

11.)

```
while ($t0 >= 12) {
  $t0 = $t0 - 1;
}
```

```
  # $t0 >= 12 is equivalent to !($t0 < 12)
  li $t1, 12
loop_begin:
  sltu $t2, $t0, $t1  # $t0 < 12?
  bne $t2, $zero, loop_end  # if !($t0 < 12), jump out of loop
  addiu $t0, $t0, -1
  j loop_begin
loop_end:
```

12.)

```
do {
  $t0 = $t0 - 1;
} while ($t0 >= 12);
```

```
  # $t0 >= 12 is equivalent to !($t0 < 12)
  li $t1, 12
loop_begin:
  addiu $t0, $t0, -1
  sltu $t2, $t0, $t1  # $t0 < 12?
  beq $t2, $zero, loop_begin # if !($t0 < 12), jump to
                             # start of loop
```