C to MIPS (If Statements) Practice Exercises Part II

Students, it is extremely important (for the learning process) for you to try and solve these questions before viewing the answers.

1) Try to translate the following to MIPS code, where i maps to $s0, and j maps to $s1 (the solution is provided at the end of the document), with the additional constraint that you can only use beq (branch on equal)

```
if (i == j)
        i = i + 1;
j = j - 1;
```

2) Try to translate the following to MIPS code, where i maps to $s0, and j maps to $s1 (the solution is provided at the end of the document), with the additional constraint that you can only use beq (branch on equal)

```
if (i == j)
        i = i + 1;
else
        j = j - 1;
j = j + i;
```

3) Try to translate the following to MIPS code, where i maps to $s0, and j maps to $s1 (the solution is provided at the end of the document), with the additional constraint that you can only use beq (branch on equal)

```
if ((i == j) && (i == k))              \\ && is the AND logical operator
        i = i + 1;
else
        j = j - 1;
j = i + k;
```

Solution to Question 1 (i maps to $s0, and j maps to $s1):

C Code:

```
if (i == j)
        i = i + 1;
j = j - 1;
```

```
beq  $s0, $s1, L1     # branch if ( i == j )
j L2
L1:
addi $s0, $s0, 1      # i = i + 1
L2
addi $s1, $s1, -1     # j = j - 1
```

Solution to Question 2 (i maps to $s0, and j maps to $s1):

C Code:

```
if (i == j)
        i = i + 1;
else
        j = j - 1;
j = j + i;
```

```
beq  $s0, $s1, IF       # branch if  ( i == j )
addi $s1, $s1, -1       # j = j - 1
j L1                    # jump over else
IF:
addi $s0, $s0, 1        # i = i + 1
L1:
add $s1, $s1, $s0       # j = j + i
```

Solution to Question 3 (i maps to $s0, j maps to $s1, and k maps to $s2):

```
if ((i == j) && (i == k))              \\ && is the AND logical operator
        i = i + 1;
else
        j = j - 1;
j = i + k;
```

```
beq  $s0, $s1, IF1   # condition 1: branch if ( i == j )
j ELSE
IF1:
beq  $s0, $s2, IF2   # condition 2: branch if ( i == k )
j ELSE
IF2:
addi $s0, $s0, 1        # if-body: i = i + 1
j L1                    # jump over else
ELSE:
addi $s1, $s1, -1       # else-body: j = j - 1
L1:
add $s1, $s0, $s2       # j = i + k
```