**Question:**

Write a function (**recitation\_six)** that takes three parameters. It should iterate from the first parameter to the second, either up or down depending on the third parameter, and return the sum. Use positive increment (+1) if the third parameter is 0. Use negative (-1) if the third parameter is 1.

**Confusion:**

Some people thought that the question is asking to write a function that takes three parameters. If third parameter is 0, they would add all the numbers from first parameter to second parameter using an incrementing loop. And if the third parameter is 1, they would decrement a loop from second parameter to first.

**If someone does this, you’d have to consider their program correct since the question is a little unclear**. Their code might look something like this:

func\_name(params):

IF\_STATEMENT(param3==0):

LOOP from param1 to param2: [INCREMENTING LOOP]

SUM(param1+…..+param2) [INCLUSIVE]

RETURN SUM

ELSE\_IF\_STATEMENT(param3==1):[

LOOP from param 2 to param 1:

SUM(param2+…..+param1) [INCLUSIVE]

RETURN SUM

INVOKE func\_name

PRINT func\_name

**In this case, their output would be the same for:**

**INPUT: (1,20,1)**

**INPUT: (1,20,0)**

**Rubric for this solution:**

Function Invocation – 5 pts

Function Header—5 pts

If elif—15 pts (partial)

Incrementing and decrementing loops—10 (partial)

Summation—10 pts (partial)

Return statements—5 pts

**Solution#2:**

Some people thought that you’d pass three parameters to the function. If your third parameter is 1, you would decrement your loop from your first parameter to second parameter. If your first parameter is 0, your loop would increment from first parameter to second parameter. **[THIS APPROACH DOESN”T WORK PROPERLY IF FIRST PARAMETER IS SMALLER THAN THE SECOND]**

**Code:**

func\_name(params):

IF\_STATEMENT(param3==0):

LOOP from param1 to param2: [INCREMENTING LOOP]

SUM(param1+…..+param2) [INCLUSIVE]

RETURN SUM

ELSE\_IF\_STATEMENT(param3==1):[

LOOP from param 1 to param 2: [DECREMENT]

SUM(param1+…..+param2) [INCLUSIVE]-**This part might result in negative values**

RETURN SUM

INVOKE func\_name

PRINT func\_name

**Rubric:**

Function Invocation – 5 pts

Function Header—5 pts

If elif—15 pts (partial)

Incrementing and decrementing loops—10 (partial)

Summation—10 pts (partial)

Return statements—5 pts

**My suggestion:**

Since Dan didn’t mention whether we should even consider cases where first parameter wouldn’t be greater than second parameter for decrement. Therefore, if someone has negative values in their output, don’t take points off for that.