Name:_____ 11/1/07

1. Translate the following high-level language code segment to MIPS assembly language. Use the registers indicated in the code.

a)
```
if X < Y then
    min = X
else
    min = Y
end if
```

b)
```
for $1 = 0 to 100 by steps of size 10 do
    if ($3 < $1) AND ($2 >= 50) then
        $2 = $2 + $3
    end if
end for
```

c)
```
while ($8 > 20) do
    if ($8 < 100)  OR ($8 > 200) then
        $7 = $8
        $8 = $8 - 10
    else
        $8 = $8 - R7
    end if
    $7 =  $6 + 4
end while
sum = 0;
```

```
integer firstUnsortedIndex, testIndex, elementToInsert;
for firstUnsortedIndex = 1 to (length-1) do
    testIndex = firstUnsortedIndex-1;
    elementToInsert = numbers[firstUnsortedIndex];
    while (testIndex >=0) AND (numbers[testIndex] > elementToInsert ) do
        numbers[ testIndex + 1 ] = numbers[ testIndex ];
        testIndex =  testIndex - 1;
    end while
    numbers[ testIndex + 1 ] = elementToInsert;
end for
```

2. Write MIPS Assembly Language code for the above insertion sort algorithm

```
            .data
numbers:    .word 20, 30, 10, 40, 50, 60, 30, 25, 10, 5
length:     .word 10

            .text
             .globl main
main:
```

```
    li      $v0, 10             # system call to exit
    syscall
```