Q1

Assumptions:

1. When function starts, the stack looks like:

---------------------- <-sp

| parameters |

---------------------- <-fp

| caller’s AR |

----------------------

1. The call is a valid call, jal doubleSum

doubleSum:

#enter

#push ra

sw $ra 0($sp) #call link

subu $sp $sp 4 #push

#push ctrl lnk

sw $fp 0($sp) #ctrl lnk

subu $sp $sp 4 #push

#set fp

addu $fp $sp 20 #8+3\*12

#load

lw $t0 -8($fp)

lw $t1 -12($fp)

lw $t2 -16($fp)

#operations

add $t2 $t0 $t1

li $v0 2

mult $v0 $v0 $t2

#exit

lw $ra -12($fp) #load return address

move $t0 $fp #save control link

lw $fp -16($fp) #restore fp

move $sp $t0 #restore sp

jr $ra

Q2

Assume a, b and c are integers.

start: li $t0 0

li $t1 1

li $t2 c

bge $t0 $t2 while

inif: bge $t1 $t2 while

inifif:

while: bge $t2 10 end

addu $t2 $t2 1

j while

end:

CFG:

../../Downloads/Untitled%20Diagram.png

Q3:

$t0 = 6

$t1 = 4

$t2 = 8

$t3 = undefined

$ra = 0

pc = 0