i: $i

j:$j

k:$k

$23 = &score[0]

#~~~~~

sll $10, $i, 2 #mul $10, $i, 4

add $t1, $23, $t0

lw $t2, ($t1)

loop:

bne $t2, $k, exit

add $i, $i, $j

j loop

exit:

better version:

sll $t0, $i, 2 #mul $10, $i, 4

add $t1, $23, $t0

lw $t2, ($t1)

loop:

bne $t2, $k, exit

add $i, $i, $j

sll $t0, $i, 2

add $t1, $23, $0

lw $t2, ($t1)

beq $t2, $k, loop

exit:

loop1:

add $i, $i, $j

sll $t0, $i, 2

add $t1, $23, $t0

lw $t2, ($t1)

beq $t2, $t0, loop1

MIPS Instruction formats:

R-format for (ALU)

32 sub

|000000| rs | rt | rd | FUNC EXT OPCODE|

|opCode| 5 | 5 | 5 | |

rd rs rt

e.g. add $13, $23, $17

I- format for (I stands for immediate)

(load/store, cond. branches, ALU w. costant)

(unique) 32 sub

|Opcode| rs | rt | rd | I |

| 6 | 5 | 5 | 5 | 16 |

e.g. beq $13, $19, exit

rs rt

J-format for