# *Início do algoritmo bubble sort*

lw 1,0,vetor # *R1 = contador i*

loop1: sw 1,0,i # *salva i na memória*

loop2: lw 2,0,j # *R2 = j*

beq 1,2,fimloop2

lw 4,2,vetor # *R4 = a*

addi 3,2,1 # *R3 = j+1*

sw 3,0,j # *salva j+1 memória*

lw 5,3,vetor # *R5 = b*

nand 6,5,5

addi 6,6,1 # *R6 = - b*

add 7,4,6 # *R7 = a - b*

lw 6,0,neg

nand 7,6,7 # *R4 = 1111 se positivo, 0111 se negativo*

lw 6,0,neg1

beq 6,7,troca

retu: beq 0,0,loop2

fimloop2: lw 1,0,i # *lê i da memória R1 = i*

addi 1,1,-1 # *i = i - 1*

sw 1,0,i # *salva i na memória*

addi 4,0,1 # *R4 testa se i = 1*

beq 1,4,fimloop1

addi 2,0,1 # *reinicia j*

sw 2,0,j # *salva j na memória*

beq 0,0,loop2 # *reinicia loop2*

fimloop1: lw 1,0,vetor # *limpa os registradores*

addi 7,0,1

movi 2,0

movi 3,0

movi 4,0

movi 5,0

movi 6,0

iniloop3: lw 2,7,vetor # *imprimirá, um por um, os termos do vetor em R2*

beq 1,7,fimalgo

addi 7,7,1

beq 0,0,iniloop3

fimalgo: halt

troca: sw 4,3,vetor

addi 3,3,-1

sw 5,3,vetor

beq 0,0,retu

######### *Inicio da pilha* #########

i: .fill # *ultimo valor de i*

j: .fill 1 # *j sempre começará em 1*

t:

neg: .fill -32768

pos: .fill 32767

neg1: .fill -1

vetor: .fill 10 # *1º termo é o tamanho do vetor*

.fill 10

.fill 7

.fill 5

.fill 9

.fill 6

.fill 4

.fill 8

.fill 3

.fill 2

.fill 1