```
#creating array and saving it to memory (you can change the values manually)
addi x1,x0,5
sw x1,4(x0)
addi x1,x0,8
sw x1,8(x0)
addi x1,x0,3
sw x1,12(x0)
addi x1,x0,1
sw x1,16(x0)
addi x1,x0,4
sw x1,20(x0)

#save number corresponding to total number of instances to x4
addi x4, x0, 5
#save number to x6 which will be deleted from x4 to check if all cells are sorted
addi x6, x0, 1

loop:
add x5, x0, x4
compare0:
#load [0] and [1] compare them
lw x1,4(x0)
lw x2,8(x0)
blt x1,x2,compare1
#if [0] > [1] swap places and remove x6 from x4 so check knows to repeat the loop
sw x2,4(x0)
sw x1,8(x0)
sub x5, x4, x6

compare1:
#compare [1] and [2]
lw x1,8(x0)
lw x2,12(x0)
blt x1,x2,compare3
sw x2,8(x0)
sw x1,12(x0)
sub x5, x4, x6

compare3:
#compare [2] and [3]
lw x1,12(x0)
lw x2,16(x0)
blt x1,x2,compare4
sw x2,12(x0)
sw x1,16(x0)
sub x5, x4, x6

compare4:
#compare [1] and [2]
lw x1,16(x0)
lw x2,20(x0)
blt x1,x2,check
sw x2,16(x0)
sw x1,20(x0)
sub x5, x4, x6
#both operations go the check method but first one skips the swapping
#if those elements are already sorted

check:
#if x4 and x5 are equal jump to exit = array is sorted, else: repeat the
loop
beq x4,x5,exit
jal ra, loop

exit:
```