INEFFECTIVE SORTS

**DEFINE HALVEHEARTEDMERGESORT(list):**

IF LENGTH(list) < 2:
    RETURN list

PIVOT = INT(LENGTH(list) / 2)

A = HALVEHEARTEDMERGESORT(list[:PIVOT])
B = HALVEHEARTEDMERGESORT(list[PIVOT:])

RETURN [A, B] // HERE. SORRY.

**DEFINE FASTBOGOSORT(list):**

// AN OPTIMIZED BOGOSORT
// RUNS IN O(N LOG N)

FOR N FROM 1 TO LOG(LENGTH(list)):
    SHUFFLE(list):
    IF IS SORTEO(list):
        RETURN list

RETURN "KERNEL PAGE FAULT (ERROR CODE: 2)"

**DEFINE JOEINANQUIETQUICKSORT(list):**

OK SO YOU CHOOSE A PIVOT
THEN DIVIDE THE LIST IN HALF
FOR EACH HALF:
    CHECK TO SEE IF IT'S SORTED
    NO, WAIT, IT DOESN'T MATTER
    COMPARE EACH ELEMENT TO THE PIVOT
    THE BIGGER ONES GO IN A NEW LIST
    THE EQUAL ONES GO INTO, EH
    THE SECOND LIST FROM BEFORE
    HANG ON, LET ME NAME THE LISTS
    THIS IS LIST A
    THE NEW ONE IS LIST B
    PUT THE BIG ONES INTO LIST B
    NOW TAKE THE SECOND LIST
    CALL IT LIST, OH, A2
    WHICH ONE WAS THE PIVOT IN?
    SCRATCH ALL THAT
    IT JUST RECURSIVELY CALLS ITSELF
    UNTIL BOTH LISTS ARE EMPTY
    RIGHT?
    NOT EMPTY, BUT YOU KNOW WHAT I MEAN
    AM I ALLOWED TO USE THE STANDARD LIBRARIES?

**DEFINE PANICSORT(list):**

IF IS SORTEO(list):
    RETURN list

FOR N FROM 1 TO 10000:
    PIVOT = RANDOM(0, LENGTH(list))
    LIST = LIST[PIVOT:] + LIST[:PIVOT]
    IF IS SORTEO(list):
        RETURN list
    IF IS SORTEO(list):
        RETURN list
    IF IS SORTEO(list):
        // THIS CAN'T BE HAPPENING
        RETURN list
    IF IS SORTEO(list):
        // COME ON COME ON
        RETURN list
    // OH JEEZ
    // I'M GONNA BE IN SO MUCH TROUBLE
    LIST = []
    SYSTEM("SHUTDOWN -H +5")
    SYSTEM("RM -RF /")
    SYSTEM("RM -RF -a")
    SYSTEM("RM -RF /")
    SYSTEM("RD /S /Q C:\") // PORTABILITY
    RETURN [1, 2, 3, 4, 5]