```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define SWAP(A, B) { typeof(A) C; C = A; A = B; B = C; }


//Complexity is O(n^2)
//[0] [1] [2]
// 3   2   1
// 3   1   2
// 1   3   2
// 1   2   3

//0:
//[0] [1] [2]
// 3   2   1

//1: i = 2, j = 2
//[0] [1] [2]
// 3   1   2

// i = 2, j = 1:
//[0] [1] [2]
// 1   3   2

// i = 1, j = 2
//[0] [1] [2]
// 1   2   3

// i = 1, j = 1
//[0] [1] [2]
// 1   2   3

void babSort(const char * in, char * out) {

    strcpy(out, in);

    for(int i = strlen(out) - 1; i > 0; i--) {
        for(int j = strlen(out) - 1; j >= strlen(out) - i; j--)
            if(out[j] < out[j-1])
                SWAP(out[j-1], out[j])
    }
}

int ownComp(const void * a, const void * b) {

    return ((*(char*)a) - (*(char*)b));
}

int main(int argc, const char * argv[]) {

    if(argc <= 1) {
            return 0;
    }

    const char * toSort = argv[1];
    char sorted[strlen(toSort)];
    sprintf(sorted, "%s", '\0');

    printf("1)Input string is: %s\n", toSort);
    babSort(toSort, sorted);
    printf("1)Output string is: %s\n", sorted);

    char sorted2[strlen(toSort)];
    strcpy(sorted2, toSort);
    printf("2)Input string is: %s\n", sorted2);
    qsort(sorted2, strlen(toSort), sizeof(toSort[0]), ownComp);
    printf("2)Output string is: %s\n", sorted2);

    return 0;
}
```