

```

1  #include <stdio.h>
2  #include <stdbool.h>
3  #include <stdlib.h>
4  #include <stdint.h>
5
6  #define TAM_LISTA 6 // para evitar magic numbers
7
8  typedef double Item; // redefinimos un tipo de dato
9
10 enum {
11     ASCENDENTE = 0,
12     DESCENDENTE = 1
13 };
14
15 void Burbuja_desc( Item lista[] , size_t num_elems){
16     bool listo = false;
17     Item tmp;
18     for(size_t i = 0; i < num_elems; i++){
19         for(size_t j = 0; j < num_elems && !listo; j++){
20             //listo = false;
21             if(lista[j] < lista[j+1]){
22                 tmp = lista[j];
23                 lista[j] = lista[j+1];
24                 lista[j+1] = tmp;
25                 listo = false;
26             }
27         }
28     }
29 }
30
31 void Burbuja_asc( Item lista[] , size_t num_elems){
32     bool listo = false;
33     Item tmp;
34     for(size_t i = 0; i < num_elems-1; i++){
35         for(size_t j = num_elems-1; j > 0 && !listo; j--){
36             // listo = true;
37             if(lista[j] < lista[j-1]){
38                 tmp = lista[j];
39                 lista[j] = lista[j-1];
40                 lista[j-1] = tmp;
41                 listo = false;
42             }
43         }
44     }
45 }
46
47 void Orden_Burbuja( Item lista[] , size_t num_elems, int direccion ){
48     if(direccion == ASCENDENTE){
49         Burbuja_asc( lista , num_elems);
50     }
51     else{
52         Burbuja_desc(lista , num_elems );
53     }
54 }
55
56 void print(Item list[]){
57     printf("\n[ ");
58     for (size_t i = 0; i < TAM_LISTA; i++) {
59         printf("%g ", list[i]);
60         if ( !(i == TAM_LISTA-1) ){
61             printf(", ");
62         }
63     }
64     printf(" ]\n");
65 }
66
67 int main(){
68     Item lista[TAM_LISTA] = {5.2,2.33,3.1,9.535,1.999,4.12};
69     // forma original

```

```

70     print(lista);
71
72     Orden_Burbuja(lista , TAM_LISTA , 0); // por defecto viene en ascendente
73 #if 0
74     short op;
75     printf("\n\n\tBienvenido al programa de ordenamiento de burbuja\n");
76     printf("\nIndica si debe ser ascendente(1) o descendente(2)\n\n\t\t->");
77     scanf("%i",&op);
78     switch (op) {
79         case 1:
80             Orden_Burbuja(lista , TAM_LISTA , 0); // el ultimo valor es la direccion (0 =
            ASCENDENTE y 1 = DESCENDENTE)
81             break;
82         case 2:
83             Orden_Burbuja(lista , TAM_LISTA , 1); // el ultimo valor es la direccion (0 =
            ASCENDENTE y 1 = DESCENDENTE)
84             break;
85         default:
86             printf("Dato Erroneo\n");
87     }
88 #endif
89     // forma final
90     print(lista);
91     return 0;
92 }
93

```