

```
1. /**
2.  * helpers.c
3.  *
4.  * Computer Science 50
5.  * Problem Set 3
6.  *
7.  * Helper functions for Problem Set 3.
8.  */
9.
10. #include <cs50.h>
11.
12. #include "helpers.h"
13.
14. /**
15.  * Returns true if value is in array of n values, else false.
16.  */
17. bool search(int value, int values[], int n)
18. {
19.     // TODO: implementar busqueda de algoritmos
20.     // Variables para buscar numeros binarios
21.     int inicia = 0;
22.     int termina = n - 1;
23.     int enmedio;
24.
25.     while (inicia <= termina)
26.     {
27.         // Encontrar el valor de mi variable de enmedio
28.         enmedio = (termina + inicia) / 2;
29.         if (values[enmedio] > value)
30.         {
31.             termina = enmedio - 1;
32.         }
33.         else if (values[enmedio] < value)
34.         {
35.             inicia = enmedio + 1;
36.         }
37.         else
38.         {
39.             return true;
40.         }
41.     }
42.     return false;
43. }
44.
45.
46.
47. /**
48.  * Sorts array of n values.
```

```
49.  */
50. void sort(int values[], int n)
51. {
52.     int min;
53.     // TODO: implement an O(n^2) sorting algorithm
54.     for (int i = 0; i < (n - 1); i++)
55.     {
56.         min = i;
57.         for (int j = i + 1; j < n; j++)
58.         {
59.             if (values[j] < values[min])
60.             {
61.                 min = j;
62.             }
63.         }
64.         // Swap
65.         if (min != i)
66.         {
67.             int temp = values[i];
68.             values[i] = values[min];
69.             values[min] = temp;
70.         }
71.     }
72.     return;
73. }
```