void swap (int \*x, int \*y) {

/\* Dados 2 parametros:

\*

\* - int posicion\_valor1

\* - int posicion\_valor2

\*

\* Respectivamente; Intercambia dos valores enteros.

\*/

int aux = \*x;

\*x = \*y;

\*y = aux;

}

int maximo(int vector[], int n) {

/\* Dado 2 parametros:

\*

\* - int vector

\* - int cantidad elementos vector

\*

\* Respectivamente; Devuelve la posición del mayor elemento del

\* arreglo, o -1 si el el vector es de largo 0.

\*/

int pos\_valor\_max = -1;

for (int i = 0; i < n; ++i){

/\* Se cambia la pos\_valor\_max en la 1 iteracion ya que -1 no es una

\* posicion valida para acceder al vector para comparar.

\*/

if (i == 0 || vector[pos\_valor\_max] < vector[i]) {

pos\_valor\_max = i;

}

}

return pos\_valor\_max;

}

int comparar(int vector1[], int n1, int vector2[], int n2) {

/\* Dado 4 parametros:

\*

\* - int vector1

\* - int cantidad elementos vector1

\* - int vector2

\* - int cantidad elementos vector2

\*

\* Respectivamente; Compara dos arreglos de longitud especificada.

\*

\* Devuelve -1 si el primer arreglo es menor que el segundo; 0 si son

\* iguales; o 1 si el primero es el mayor.

\*

\* Un arreglo es menor a otro cuando al compararlos elemento a elemento,

\* el primer elemento en el que difieren no existe o es menor.

\*/

int max\_iteraciones = n1;

if (n2 > n1) {

max\_iteraciones = n2;

}

for (int i = 0; i < max\_iteraciones; ++i){

/\* En el segundo termino se verifica que alla un elemento en

\* vector2 para hacer la comparacion.

\*/

if (n1 <= i || (n2 > i && vector1[i] < vector2[i])){

return -1;

}

else if (n2 <= i || vector1[i] > vector2[i]){

return 1;

}

}

/\* Si sale del ciclo es porque los vectores son iguales en elementos

\* y cantidad de elementos.

\*/

return 0;

}

void seleccion(int vector[], int n) {

/\* Dado 2 parametros:

\*

\* - int vector

\* - int cantidad elementos vector

\*

\* respectivamente, ordena el vector in place.

\*/

for (int i = n; i > 0; --i){

int pos\_max = maximo(vector, i);

swap(&vector[pos\_max], &vector[i-1]);

}

}