Sorting :

| 3 | 7 | 5 | 11 | 2 | | |

$\rightarrow$ | 2 | 3 | 5 | 7 | 11 |

void sort (vector <int> & V);

Idea: "incrementally" sort using the
following invariant :

At the start of iteration, assume
V[0...i-1]   are i smallest elements of V
in sorted order. (So, the job for
iteration i is to find the
right value for V[i].

Example: | 3 7 5 11 2   "selection
                                  Sort"
          2 | 7 5 11 3
          2 3 | 5 11 7

          2 3 5 7 | 11

How to implement iteration i ?
Then V[0...i-1] already in their
right places. Need to find the
right thing for V[i].
   V[i] should be   min {V[i],...,V[n-1]}

                    const
int indexOfSmallest (vector <int> & V, int start) {

```
// find index of minimal V[start ... n-1]
    int small index = start;
    // index of the smallest value in V[start..i]

       for (int i = start+1; i < n; i++) {
            if (V[i] < V[small index])
                  small index = i;
       }
       return small index;
    } // end of index Of Smallest.
```

Given that index Of Smallest works, how to
finish the procedure?
        the bar |
                ↓
```
       for (int i = 0; i < n-1; i++) {
       // swap V[i] ⟷ V[index Of Smallest(V, i)]
```