Quicksort: Classical imperative version:

```
procedure qsort(l,r: index);
var i,j: index; x,w: item
begin
   i := l;  j := r;
   x := a[(l+r) div 2];
   repeat
      while a[i] < x do i := i+1;
      while x < a[j] do j := j-1;
      if i <= j then
      begin w := a[i]; a[i] := a[j]; a[j] := w;
         i := i+1; j := j-1
      end
   until i > j;
   if l < j then qsort(l,j);
   if i < r then qsort(i,r);
end
```

Quicksort: Classical imperative version:

```
procedure qsort(l,r: index);
var i,j: index; x,w: item
begin
   i := l;  j := r;
   x := a[(l+r) div 2];
   repeat
      while a[i] < x do i := i+1;
      while x < a[j] do j := j-1;
      if i <= j then
      begin w := a[i]; a[i] := a[j]; a[j] := w;
         i := i+1; j := j-1
      end
   until i > j;
   if l < j then qsort(l,j);
   if i < r then qsort(i,r);
end
```

Declarative version:

```
qsort [] = []
qsort (x:l) =
    qsort (filter (<x) l)
    ++ [x]
    ++ qsort (filter (>=x) l)
```