1. Insertion sort:

Input: Possibly unsorted array of data.

Output: Sorted array of data.

```
Begin
    For i = 1 to length(array-1)
    Begin
        tmp = array[i];
        j = i – 1;
        while j >= 0 && array[j] > tmp
        Begin
            array[j + 1] = array[j];
            j--;
        End
        Array[j + 1] = tmp;
        Print the contents of the partly sorted array as a semi colon delimited list ending with a newline.
    End
End
```

2. Selection Sort:

Input: Possibly unsorted array of data.

Output: Sorted array of data.

```
Begin
    For i = 0 to length(array)-1
    Begin
        Select the smallest element from array[i] to array[length(array)-1];
        Swap the smallest element with array[i];
        Print the contents of the partly sorted array as a semi colon delimited list ending with a newline.
    End
End
```

3. Bubble Sort

Input: Possibly unsorted array of data.

ascending

Output: Sorted array of data.

```
Begin
    For i = 0 to length(array)-2
    Begin
        For j = length(array)-1 down to i+1
        Begin
            If elements in position j and j-1 are out of order, swap them
        End
        Print the contents of the partly sorted array as a semi colon delimited list ending with a newline.
    End
End
```

4. Comb Sort

Input: Possibly unsorted array of data.

Output: Sorted array of data.

```
Begin
    gap = length(array)
    swapped = false;
    while gap > 1 or swapped
    Begin
        Swapped = false
        If gap > 1
        Begin
            gap = floor(gap/1.3)
        End
        i = 0
        while i + gap < length(array)
        Begin
            if array[i] > array[i + gap]
            Begin
                swap array[i] with array[i + gap]
                set swapped to true.
            End
            i += 1
        End
        Print the contents of the partly sorted array as a semi colon delimited list ending with a newline.
        Print the gap as follows: Gap: {gap}\n
            //So if gap = 4 then the output should be: Gap: 4\n
    End
End
```