In this question you need to write a program that sorts a given fixed-length array ‘num1’ in non-decreasing order. For this you are given a different template code which is pre-initialized with 10 different values from 8th line to 18th line as shown below.

```
num1:
.word 0x03 = 3
.word 0x04 = 4
.word 0x01 = 1
.word 0x7FFFFFFE =2147483646
.word 0x80000000 = -2147483648
.word 0xFF =255
.word 0xEF = 239
.word 0xFFFFFFFF = -1
.word 0x07 =7
.word 0x09 =9
```

So, changing these values will change the contents of the pre-initialized array.

In main code you can see 2 instructions 'movw' and 'movt' written for you which loads the array address of array (also the address of the 1st element (0x03)) into register 'r4'. The last instruction will just load a constant value ‘10’ into r5, the constant length of the array.

```
movw r4, #:lower16:num1
movt r4, #:upper16:num1
mov r5, #10
```

Un- commenting the following 4 lines will print ‘3’ and ‘4’ the 1st and 2nd elements of array.

```
ldr r1, [r4,#0]
bl myPrint
ldr r1, [r4, #4]
bl myPrint
```

You algorithm should sort the array and print them in non-decreasing order at the end of the program. Notice that your algorithm should work with any valid set of pre-initialized array values which will be given as the test cases.