```perl
 1      ## Assignment: Project Four - Multi-dimensional arrays & sorting
 2      ## Author: Howard Bates (hbates@northmen.org)
 3      ## Version: 02.28.2013.01
 4      ## Purpose: Array & sorting example
 5
 6      use 5.14.2;
 7      use warnings;
 8      no warnings 'uninitialized';
 9
10      my (@arrayNumbers,@names);
11      my ($numNumbers,$continue,$counter);
12      my $YES = 1;
13
14      sub main {
15              $continue = $YES;
16              $counter = 0;
17              while ($continue == $YES) {
18                      setNumNumbers();
19                      populateArrayNumbers();
20                      populateNames();
21                      sortArrayNumbers();
22                      printArrayNumbers();
23                      printNames();
24                      setContinue();
25              }
26      }
27
28      main();
29
30      sub setContinue {
31              print "\nDo you want to continue? (0=no, 1=yes)";
32              chomp ($continue = <STDIN>);
33      }
34
35      sub setNumNumbers {
36              print "\n\nHow many numbers are you entering (no more than ten)? ";
37              chomp($numNumbers = <STDIN>);
38              validateInput();
39      }
40
41      sub populateArrayNumbers {
42              @arrayNumbers = ();
43              for (my $i = 0; $i < $numNumbers; $i++) {
44                      print "\nPlease enter a last number: ";
45                      chomp($arrayNumbers[$i] = <STDIN>);
46              }
47      }
48
49      sub sortArrayNumbers {
50              my $index = 0;
51              my $didSwap = 1;
52              my $temp = 0;
53              my $size = @arrayNumbers;
54              while ($didSwap == 1) {
55                      $index = 0;
56                      $didSwap = 0;
57                      while ($index < $size) {
58                              if ($arrayNumbers[$index] > $arrayNumbers[$index + 1]) {
59                                      $temp = $arrayNumbers[$index];
60                                      $arrayNumbers[$index] = $arrayNumbers[$index + 1];
61                                      $arrayNumbers[$index + 1] = $temp;
62                                      $didSwap = 1;
63                              }
64                              $index++;
65                      }
```

```perl
 66                 }
 67         }
 68
 69         sub printArrayNumbers {
 70                 foreach my $item (@arrayNumbers) {
 71                         print "\n$item\n";
 72                 }
 73         }
 74
 75         sub populateNames {
 76                 @names = ();
 77                 for (my $i = 0; $i < $numNumbers; $i++) {
 78                         print "\nPlease enter a first name: ";
 79                         chomp($names[$i][$i] = <STDIN>);
 80                         print "\nPlease enter a last name: ";
 81                         chomp($names[$i][$i + 1] = <STDIN>);
 82                 }
 83         }
 84
 85         sub printNames {
 86                 my $size = @names;
 87                 for (my $i = 0; $i < $size; $i++) {
 88                         print "\n$names[$i][$i] $names[$i][$i + 1]\n";
 89                 }
 90         }
 91
 92         sub validateInput {
 93                 my $MIN_NUMBERS = 1;
 94                 my $MAX_NUMBERS = 10;
 95                 my $MAX_VALIDATE = 5;
 96                 if ($counter <= $MAX_VALIDATE) {
 97                         $counter++;
 98                         if ($numNumbers < $MIN_NUMBERS || $numNumbers > $MAX_NUMBERS) {
 99                                 print "\nYou must enter a number between 1 and 10. Try again\n";
100                                 sleep 1;
101                                 setNumNumbers();
102                         }
103                 } else {
104                         die "\n\n\tToo many attempts, goodbye.";
105                 }
106         }
```