Sorting

Rather than physically rearranging an array for sorting purposes, it is preferable to sort it passively (sort an array that hold its indices) because this allows us to sort parallel arrays. Here is an example:

```
program sort
implicit none
real*8 xAR(1000), yAR(1000)
integer*4 n, actualSize, maxSize /1000/, permute(1000)
integer*4 status, err
open (unit = 1, file = 'data.txt')

n = 1
do while (.true.)
   read(1,*, IOSTAT=status) xAR(n), yAR(n)
   if (status .ne. 0) then
      exit
   end if
   n = n + 1
end do
close(unit = 1)
actualSize = n - 1

print*, "Before: ", actualSize, xAR(1), yAR(1)

call dpSort(xAR, actualSize, permute, 1, err)
call dpPerm(xAR, actualSize, permute, err)
call dpPerm(yAR, actualSize, permute, err)

print*, err
print*, "After:  ", actualSize, xAR(1), yAR(1), permute(1)

end
```

data.txt

```
12.89   56
6       8
1       67
13      5
-4      -44
23      6
```

Output:

```
Before:  6  12.89  56.
 0
After:   6 -4. -44. 5
```