

## Assignment 5 1 (method 1) - Program

```
1 import numpy
2
3 #read data into an array
4 marks = numpy.loadtxt('sample_marks.dat')
5
6 #initialise all statistics with relevant initial values
7 maxmark = -1.
8 minmark = 101.
9 maxloc = -1.
10 minloc = -1.
11 meanmark = 0.
12
13 #go through each index/mark pair (the 'enumerate' function on an array,
14     returns pairs of values of indices and the actual array values)
15 for i,mark in enumerate(marks):
16     meanmark += mark #keep a running tally of the sum of the marks
17     if (mark > maxmark):
18         #if the current mark exceeds the current maximum, replace the max and
19         position of the max
20         maxmark = mark
21         maxloc = i
22     if (mark < minmark):
23         #same for minimum
24         minmark = mark
25         minloc = i
26
27 #divide the total sum by the number of elements in the array to get the mean
28 meanmark /= float(len(marks))
29
30 #create a formatted line with the relevant numbers inserted
31 output = 'Maximum mark was %d for student %d\nMinimum mark was %d for student
32         %d\nMean mark was %f'%(maxmark,maxloc,minmark,minloc,meanmark)
33
34 print output #print it to screen
35 #write it to file
36 fout = open('assign_5_1_method1.out','w')
37 fout.write(output+'\n')
38 fout.close()
```

## Assignment 5 1 (method 1) - Output

```
1 Maximum mark was 97 for student 26
2 Minimum mark was 5 for student 17
3 Mean mark was 58.288889
```

## Assignment 5 1 (method 2) - Program

```
1 # quick (and cheating, arguably; given the specification of the assignment)
2   version of the assign_5_1 code. Note that similar can be done in fortran;
3   this just illustrates exactly how compact a python code can be for a given
4   task, if there are already external routines in modules for it. Keep in
5   mind that the *true* length of the code actually being executed is
6   substantially longer; as we do not see, here, all of the numpy code which
7   actually does all of this
8
9 import numpy
10
11
```

```

4 #read in
5 marks = numpy.loadtxt('sample_marks.dat')
6
7 #create line with insertions for relevant values; calculate them in-place;
  using numpy routines
8 output = 'Maximum mark was %d for student %d\nMinimum mark was %d for student
  %d\nMean mark was %f'%(numpy.max(marks),numpy.argmax(marks),numpy.min(
  marks),numpy.argmin(marks),numpy.mean(marks))
9
10 #output it all
11 print output
12 fout = open('assign_5_1_method2.out','w')
13 fout.write(output+'\n')
14 fout.close()

```

### Assignment 5 1 (method 2) - Output

```

1 Maximum mark was 97 for student 26
2 Minimum mark was 5 for student 17
3 Mean mark was 58.288889

```

### Assignment 5 2 - Program

```

1 #initialise the array
2 myarray = [5,7,3,1,5,6]
3
4 #loop for number of whole passes
5 for i in range(len(myarray)-1):
6     switched = False #initialise each pass
7     #loop performing one pass of the array
8     for j in range(len(myarray)-i-1):
9         #if the array elements are out of the expected order...
10         if (myarray[j] > myarray[j+1]):
11             #...switch them over
12             t = myarray[j]
13             myarray[j] = myarray[j+1]
14             myarray[j+1] = t
15             #set to 'True' to indicate this pass has resulted in at least one
  switch, and more passes may still be needed to sort the data
16             switched = True
17     if (not switched):
18         #jump out of 'i' loop if no switches occurred on mst recent pass
19         break
20
21 #output the array to the screen...
22 print myarray
23 #...and to the file
24 fout = open('assign_5_2.out','w')
25 fout.write(str(myarray)+'\n')
26 fout.close()

```

### Assignment 5 2 - Output

```

1 [1, 3, 5, 5, 6, 7]

```

## Assignment 5 3 - Program

```
1 import numpy
2
3 #read in array from file
4 myarray = numpy.loadtxt('sample_random.dat')
5
6 #loop for number of whole passes
7 for i in range(len(myarray)-1):
8     switched = False #initialise each pass
9     #loop performing one pass of the array
10    for j in range(len(myarray)-i-1):
11        #if the array elements are out of the expected order...
12        if (myarray[j] > myarray[j+1]):
13            #...switch them over
14            t = myarray[j]
15            myarray[j] = myarray[j+1]
16            myarray[j+1] = t
17            #set to 'True' to indicate this pass has resulted in at least one
            #switch, and more passes may still be needed to sort the data
18            switched = True
19    if (not switched):
20        #jump out of 'i' loop if no switches occurred on mst recent pass
21        break
22
23 #output the array to the screen...
24 print myarray[0:10] #first 10 elements
25 print myarray[-10:] #final 10 elements, the '-10' means '10 from the end of
    the array', the colon without a trailing value mean 'onwards'; i.e. 'ten
    from the end, onwards'
26 #...and to the file
27 fout = open('assign_5_3.out','w')
28 fout.write(str(myarray[0:10])+'\n'+str(myarray[-10:]))+'\n')
29 fout.close()
```

## Assignment 5 3 - Output

```
1 [ 17.  25.  30.  34.  35.  37.  37.  39.  40.  40.]
2 [ 833.  857.  911.  924.  936.  936. 1149. 1204. 1285. 1335.]
```