Def print.sum(1st)

“””

>>>print\_sum([1, 2, 3, 4, 5, 6])

21

>>>print\_sum([500, -1, 2.5, 7])

508.5

“””

print(sum(1st))

Def new\_min(1st)

“””

‘Returns least number.’

>>> new\_min([5, 6, 10, 4, 20, 3])

3

“””

least = 1st[0]

for item in list[1:]:

if item < least:

least = item

return least

Figure 1 This is what min does

Notes: [3:] is the same as: 3 - len

[:3] is the same as: +3

**How to find the indices of the two smallest values in a list:**

Def find\_two\_smallest(L):

“””(list of floats) -> tuple of (int, int)

Return a tuple of the indices of the two smallest values in list L.

>>>find\_two\_smallest([809, 834, 477, 478, 307, 122, 96, 102, 324, 476])

(6, 7)

“””

**# Find the index of the minimum and remove that item**

smallest = min(L) **All on page 226**

min1 = L.index(smallest)

L.remove(smallest)

**#Find the index of the new minimum**

next\_smallest = min(L)

min2 = L.index(next\_smallest)

**#Put smallest back into L**

L.insert(min1, smallest)

**#Fix min2 in case it was affected by the reinsertion**

if min1 <= min2:

min2 += 1

return(min1, min2)

[[name, [earned, total]], [name, [earned, total], …]]

{name1: (earned, total),

name2: (earned, total),

……..}

Dictionary notes:

[days

[student

[grades]

]

]

{day: 90

student\_name: {grade\_name: val 1, val 2, val3

}

}