2-15)

Passing the parameters by reference leads to the value of 2 being assigned to M. Passing by value-return results in M = 3.

2-16)

In both value-return cases, the parameters (i, A[i]) are passed into the subroutine, where i = 1, and A[i] =A[1] = 10. Both variants will print the second parameters value (10) immediately.

a) In the instance of a single address calculation, K, which references i, is set to 2 and X, which references A[1], is set to 20. When the subroutine completes it returns both of these values to the referenced variables and prints A[1], which is now 20, and A[2], which remains 11.

Thus in this instance the program prints 10, 20, 11

b) If addresses are calculated at both entry and exit, where the results from the formal K is stored in ‘I’ before recomputing the address of A(i):

The Caller: Call SUB(I, A(I)), then I = 1 and A(1) = 10
The Callee: formal/actual will substitute 1 into its proper K, X = 10
output 10
then K will be set to 2, and X will have 20.
return
The Caller: I will still have 2, A(I) => A(2) which is 20
A(1) will remain 10
Print A(1), A(2) => 10, 20
This instance will print 10,10,20

c) If addresses are calculated at both entry and exit, where the results from the formal K is recomputed before storing in ‘I’:

The Caller: Call SUB(I, A(I)), then I is 1, and A(1) is 10
The Callee: formal/actual substitution will substitute 1 into the proper K, X = 10
output 10
K will be set to 2, and X will have 20
return
The Caller: A(I) is computed, where I = 1 still
A(1) is now 20, A(2) is still 11
Print A(1), A(2)
This instance will print 10, 20, 11